

BAKALÁŘSKÁ PRÁCE

Databáze kabelových rozvodů na VŠB-TUO
The Database of Cable Distribution at VSB-TUO

Zadání bakalářské práce

Student:

Petr Holas

Studijní program:

B2645 Elektrotechnika, sdělovací a výpočetní technika

Studijní obor:

2612R018 Elektronika a sdělovací technika

Téma:

Databáze kabelových rozvodů na VŠB-TUO.
The Database of Cable Distribution at VSB-TUO.

Zásady pro vypracování:

1. Linux, Apache, MySQL a PHP.
2. Požadavky na evidenci kabelových rozvodů.
3. Realizace databáze pro evidenci kabeláže s konfigurací přes web.
4. Vytvoření příručky pro instalaci, administraci a užívání databáze.

Seznam doporučené odborné literatury:


- KOSEK, J. *XML a PHP 5*. Praha : Grada, 2004. 492 s. ISBN 80-247-1116-8.
KOSEK, J. *PHP Tvorba interaktivních aplikací*. Praha : Grada, 1999. 492 s. ISBN 80-7169-373-1.
WILLIAMS, H., LANE, D. *PHP a MySQL*. Praha: Computer Press, 2002. ISBN 80-7226-760-4.

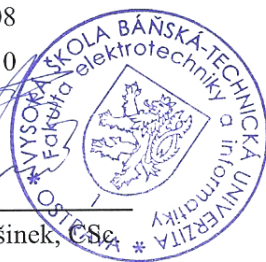
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2010


prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry




prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě dne _____

Podpis _____

Poděkování

Chtěl bych především poděkovat Doc. Ing. Miroslavovi Vozňákovi, Ph.D., vedoucímu bakalářské práce, za všechny podněty, vstřícnost a ochotu. Také děkuji své rodině a manželce Silvii za podporu a toleranci při psaní této práce.

Abstrakt

Bakalářská práce popisuje tvorbu webové aplikace. Na začátku popisují databázi, software a programovací jazyky, které můžeme použít při tvorbě webové aplikace. Následuje popis instalace vybraného softwaru, který byl použit pro vytvoření a odzkoušení fungování webové aplikace. Dále uvádím seznam všech vytvořených skriptu v PHP kódu a popis toho, co jednotlivé skripty provádějí. Jako příloha byla vytvořena příručka na obsluhu webové aplikace pod názvem „Příručka pro ovládání kabelové knihy“.

Abstract

The bachelor work describes creation of web application. First, I describe database, software and programme languages, than we can use for creation of web application. Than follows description of installation of chosen software which was used for creation and testing of function of web application. Than I give the list of all created scripts in the PHP code and description how the scripts practise. As enclosure the manual for using web application was created under the name „Manual for using the cabel book“.

Klíčová slova

Databáze; PHP

Key words

The Database; PHP

Obsah

1	Úvod.....	4
2	Databáze.....	4
	2.1 Systém řízení dat bází dat.....	4
	2.2 Databázový systém	5
3	Jazyk SQL.....	8
	3.1 Jazyk pro definici dat	9
	3.2 Jazyk pro manipulaci dat	9
4	Popis PHP	9
	4.1 Historie PHP	10
	4.2 Výhody PHP.....	10
	4.3 Nevýhody PHP.....	10
5	Jazyk HTML	11
	5.1 Struktura dokumentu	11
	5.2 Druhy značek	12
6	Open Source.....	13
7	Linux	14
	7.1 Vývoj Linuxu	15
	7.1.1 Distribuce	16
	7.1.1.1 Binární	16
	7.1.1.2 Zdrojové.....	18
	7.1.1.3 Live.....	18
8	Apache.....	19
9	PhpMyAdmin	19
	9.1 Souhrn schopností phpMyAdmin	20
10	Virtuální počítač.....	21
	10.1 Hardwarový virtuální stroj	21
	10.2 Aplikační virtuální stroj.....	22
	10.3 Virtuální prostředí.....	22

11	VMware.....	22
12	Realizace.....	23
12.1	Instalace virtuálního počítače.....	23
12.2	Instalace operačního systému	24
12.3	Instalace LAMP	26
12.3.1	Apache moduly	28
12.3.2	PHP moduly	28
12.3.3	PhpMyAdmin	29
12.4	Ověření funkčnosti instalovaného softwaru.....	29
12.4.1	Ověření Apache serveru	29
12.4.2	Ověření MySQL.....	30
12.4.3	Ověření PHP.....	30
12.5	Vytvoření webové aplikace databáze.....	31
12.5.1	Vytvoření databáze a tabulek	32
12.5.1.1	Vytvoření tabulek	32
12.5.1.2	Propojení tabulek	34
12.5.2	Tvorba skriptu webové aplikace	34
12.5.2.1	Skripty určené pro část Telefonní ústředny	36
12.5.2.2	Skripty určené pro část kabel.....	39
12.5.2.3	Skripty určené pro část servis	42
12.5.2.4	Ostatní skripty	44
13	Závěr.....	46
14	Seznam použité literatury.....	47
15	Seznam obrázků	48
16	Seznam příloh	48

1 Úvod

Cílem této bakalářské práce je vytvoření webové aplikace pro databázi kabelových rozvodů na VŠB-TU Ostrava. Při vytváření webové aplikace používat doporučený software, jako je operační systém LINUX, server Apache, databázový server MySQL a skriptovací jazyk PHP. Veškerý tento software je open source, takže je přístupný všem, kteří mají zájem studovat a rozšiřovat tuto webovou aplikaci. Nejdříve krátký úvod co je to databáze, jaké databázové systémy máme k výběru. Něco o databázovém jazyku SQL a skriptovacím jazyku PHP. Vysvětlení co znamená označení open source software a přehled operačního systému Linux. Následuje krátký popis dalšího použitého softwaru.

V další části realizace je uveden postup instalace veškerého softwaru, který jsem používal při vytváření skriptů webové aplikace a jejich následném odzkoušení. Při psaní skriptů ve skriptovacím jazyku PHP jsem používal textový editor gedit, který je obsažen v operačním systému Ubuntu. Na vytvoření tabulek jsem používal phpMyAdmin, což je webová aplikace napsaná v PHP, která poskytuje kompletní webové rozhraní pro administraci MySQL. V další části je seznam všech vytvořených skriptů a krátký popis, co jednotlivé skripty dělají.

2 Databáze

Teorie databázových systémů se začala rozvíjet už v první polovině minulého století, ale v takové podobě jak jí známe dnes, byla definována až v sedmdesátých letech. I když se to na první pohled nezdá, jedná se o poměrně složitou teorii.

Databázi můžeme chápat jako úložiště údajů, které jsou uloženy a zpracovány nezávisle na aplikačních programech. Databáze zapouzdřují jednak vlastní údaje, ale také relační vztahy mezi jednotlivými prvky a objekty v databázi, schémata popisující struktury údajů a integritní omezení.

2.1 Systém řízení dat bází dat

Speciální software pro přístup k údajům v databázi nazýváme česky systém řízení báze dat (SRBD), anebo anglicky Database Management Systém (DBMS). Uživatel, případně aplikační program potom nemusí znát fyzickou strukturu uložení údajů, protože k údajům v databázi přistupují prostřednictvím systému řízení dat. Komunikace klienta nebo [7]

aplikačního programu SŘBD probíhá pomocí jazyka SQL. Pro moderní SŘBD jsou typické některé charakteristické rysy, jako například:

Transakční zpracování znamená, že složitější manipulace s údaji, která se skládá z posloupnosti určitých kroků, je vykonávána jako transakce, která převede databázi z jednoho konzistentního stavu do druhého. Zjednodušeně řečeno, buď všechny operace v transakci proběhnou úspěšně, nebo neproběhnou vůbec, tedy databáze bude uvedena do konzistentního stavu, v jakém byla před transakcí.

Zotavení se z chyb a nehod. Když při zpracování údajů dojde k chybě a k tomu ve skutečnosti nejen při transakcích může dojít, systém se musí z této chyby zotavit. Když vlivem nějaké nepředvídané události, například havárie nebo živelné pohromy, dojde ke ztrátě údajů, měl by se systém po těchto událostech zotavit a pokračovat v práci například se záložní databází a podobně.

Víceuživatelský přístup předpokládá efektivní řízení přístupu k údajům ze strany více uživatelů nebo klientských aplikací. SŘBD proto musí umět definovat přístupová práva jednotlivých uživatelů k databázovým objektům a specifikovat rozsah jejich oprávnění. Někteří uživatelé mohou údaje do databáze zapisovat, případně je mazat, jiní mají zpřístupněné jen čtení údajů.

Ochrana údajů, tento pojem lze definovat jako ochranu před ztrátou údajů, například při haváriích hardwaru a podobně, ale můžeme ho také definovat jako ochranu údajů před jejich možnou krádeží, nebo zneužitím. Případně jako nutnost zabránění přístupu neoprávněných osob k údajům v databázích.

Architektura. V současnosti při velkých objemech uchovávaných a zpracovávaných údajů mohou být buď uloženy centralizovaně na jednom serveru, nebo distribuovaně na více serverech.

2.2 Databázový systém

Tento pojem zapouzdřuje jednak údaje, které jsou uloženy a spravovány v databázi, a také i software pro přístup k těmto údajům. Souhrn nástrojů, postupů a technik, které se využívají v souvislosti s databázemi, nazýváme databázové technologie. Databázový systém je tedy tvořen systémem řízení báze dat (SŘBD) a databází. [7]

Databázové systémy mohou být:

- hierarchické a síťové databázové systémy jsou aplikační programy závislé na databázi, z čehož vyplývá například problematická údržba a podobně,
- relační databáze je typická neprocedurální manipulace s daty, ukládání jednoduchých dat s pevnou strukturou, tedy v tabulkové formě,
- objektové databázové systémy používají složité datové struktury a složitá pravidla založená na obchodní logice. [7]

Na výběr máme z velkého množství databázových systémů například:

Databázový systém Oracle - Je považována za typického představitele relačních databázových systémů. Do této databáze se dají ukládat dokumenty v různých formátech (HTML, text, office dokumenty) a vyhledávat v jejich obsahu. Součástí této databáze je komponenta HTML DB, umožňující tvorbu webových aplikací v prostředí webového prohlížeče. Databáze obsahuje komponenty Workflow, která se dá využít při práci s dokumenty, relačními daty, řešení schvalovacích procesů v aplikacích i při integraci aplikací. V databázi Oracle je uloženo až 80% geopolohových dat, lze tam ukládat a analyzovat údaje např. o rozmístění zákazníků, poloze a půdorysu budov.

Databáze PostgreSQL - zjednodušeně Postgres, je objektově – relační databázový systém. Jedná se o free a open source software. Je primárně vyvíjen pro Linux, resp. unixové systémy. Vyniká stabilitou, rychlostí, dobrou podporou a dobrou integrací pokročilých technologií.

MSSQL - Microsoft SQL Server je relační databázový systém programovaný firmou Microsoft. Programovacími jazyky jsou SQL a T-SQL.

SQLite - je relační databázový systém, obsažený v relativně malé knihovně napsané v jazyce C. SQLite je knihovna, která se přilinkuje k aplikaci a pomocí jednoduchého rozhraní ji lze začít využívat. Každá databáze je uložena v samostatném souboru .dbm (database manager), kde data se ukládají pomocí jednoduchého primárního klíče do stejně velkých bloků a používá haschovací techniky pro přístup k datům při vyhledávání podle určitého klíče. Formát databázových souborů je nezávislý na operačním systému a SQLite tak představuje nástroj pro přenos strukturovaných dat.

Informix - jedná se o databázový server určený především pro střední a menší podniky.

Firebird - jedná se o multiplatformní relační databázi. Mezi její výhody patří především široká podpora ze strany vývojových nástrojů, výhodou je i jednoduché zálohování. Vhodná pro firmy střední velikosti.

Filemaker - je to multiplatformní relační databáze, která umožní upravit data „přetažením“ nového údaje např. do formuláře, na obrazovku.

Ingres - je relační databázový systém s komerční podporou, který je také šířen pod open source licencí.

Teradata - relační databáze, která může přijmout velké množství souběžných požadavků z klientských aplikací.

DB2 - IBM Database 2- relační databáze firmy IBM, založena na dotazovacím jazyku SQL.

Sybase - tato databáze je především využívána ve zdravotnictví a pro investiční trhy. Nabízí verze pro Windows, pro Linux.

MySQL - je databázový systém, komunikující pomocí jazyka SQL. Díky snadné implementovatelnosti, výkonu a volné šířitelnosti má vysoký podíl na v současné době používaných databázích. Velmi častá je kombinace Linux, MySQL, Apache a PHP jako základního softwaru webového serveru.

Databázový systém je založen na tabulkách. Řádky zastupují jednotlivé záznamy, sloupce pak obsahují jednotlivé vlastnosti.

MySQL - přehled podporovaných vlastností:

- Cizí klíče,
- Transakce,
- Podpora různých znakových sad a časových pásem v datech,
- Poddotazy,
- Uložené procedury,
- Tiggery,
- Pohledy. [8]

MySQL nabízí několik typů databázových tabulek (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů:

- MyISAM – nejpoužívanější, bez podpory transakcí,
- InnoDB – podpora transakcí,
- BerkeleyBD (BDB),
- MEMORY – práci s daty v paměti,
- NDB Cluster – úložiště pro clusterové databáze (od verze 5.0),
- ARCHIVE – komprimované tabulky, bez podpory indexů,
- CSV – ukládání dat v prostých textových souborech.

Výhody

MySQL je optimalizováno již od svého počátku pro vysokou rychlost. Je nabízeno pod open source licenci, čímž je snadno dostupné. Téměř každý webhosting nabízí k využití tuto databázi.

Nevýhody

Optimalizace rychlosti byla provedena na úkor některých zjednodušení. Dále u sloupců, které mají číselný formát by neměly figurovat uvozovky či apostrofy. Mohou se pak objevit potencionálně zneužitelné chyby, velmi častým případem je SQL injection. [8]

3 Jazyk SQL

Historie databázového jazyka SQL (tehdy se tak ještě nejmenoval) začíná v roce 1974, kdy byly uveřejněny první práce Dr. Codda o relačních databázových modelech. První použitelnou verzi databázového jazyka vytvořila firma IBM pod názvem SEQUEL (Structured English Query Language). Už tehdy byla patrná snaha vytvořit databázový jazyk, který by vycházel z přirozeného jazyka – angličtiny. Postupně se k tomuto standardu přidávaly další firmy (Oracle, SyBase, Informix), a tak vznikl „nepsaný standard“ databázového jazyka s názvem SQL. Postupně byly přijaty vylepšené a upravené standardy jazyka SQL s názvem SQL-86 a dále SQL-92. Pro verzi SQL-92 se vžil zkrácený název SQL-2. Ve vývoji je už nová verze s pracovním názvem SQL-3, a dokonce už byly zahájeny přípravné práce na normě SQL-4. Jazyk SQL můžeme použít jednak jako dotazovací jazyk pro práci s údaji v relační databázi, případně jako část hostitelského jazyka pro vývoj databázových aplikací. Například databázová platforma Oracle má implementovaný [7]

procedurální jazyk PL/SQL, anebo Microsoft SQL Server 2000 má implementovaný jazyk Transact SQL (T-SQL). Jazyk SQL můžeme rozdělit na dvě základní podmnožiny.

3.1 Jazyk pro definici dat

Umožňuje definovat struktury a vytvářet v databázi objekty, jako tabulky, pohledy, indexy a podobně, případně měnit jejich strukturu anebo je rušit (odstraňovat). Na tomto místě je také potřeba si uvědomit rozdíl mezi vymazáním a zrušením. Vymazat můžeme například všechny, nebo jen některé údaje z databázové tabulky, přičemž tato tabulka jako objekt zůstane zachována a připravená pro případné vložení nových údajů. Když ale tabulku zrušíme (odstraníme) například pomocí příkazu DROP TABLE, bude z databáze odstraněna databázová tabulka současně se všemi údaji a objekty, které obsahuje natrvalo.

Do této skupiny patří například příkazy:

CREATE DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE,
CREATE INDEX, DROP INDEX, CREATE VIEW, ALTER VIEW,
DROP VIEW, DROP INDEX, CREATE SEQUENCE, ALTER SEQUENCE.

3.2 Jazyk pro manipulaci dat

Umožňuje manipulaci s údaji, to znamená výběr a vkládání údajů a jejich aktualizaci, vymazání údajů a samozřejmě velmi mocný příkaz SELECT pro výběr údajů.

Patří sem čtyři hlavní příkazy:

SELECT, UPDATE, INSERT, DELETE. [7]

4 Popis PHP

PHP je programovací jazyk, který je určený především pro programování internetových stránek. PHP skripty jsou prováděny především na straně serveru, uživatel uvidí výsledek jejich činnosti na HTML stránce. Syntaxe jazyka je inspirována jinými programovacími jazyky např. Perl, C, Pascal, Java. Umožňuje vytvářet a upravovat grafické soubory, dokumenty, umožňuje práci se soubory, přístup k databázovým systémům (MySQL, ODBC, Oracle, MSSQL), podporuje řadu internetových protokolů, např. HTTP, SMTP, FTP, IMAP. [10]

4.1 Historie PHP

PHP bylo označení pro Personal Home Page, tedy osobní domácí stránky. V roce 1994 binární část Common Gateway Interface napsal dánský programátor Rasmus Lerdorf v programovacím jazyku C. Poté jej skloubil s jiným programem, který sám napsal. Po spojení s Form Interpreter tak vznikla kombinace PHP/FI, obsahující širokou implementaci pro programovací jazyk C a tato verze umožnila komunikaci s databázemi a byla možná tvorba prvních jednoduchých webových aplikací. Veřejně bylo vydáno PHP 8. Června 1995.

Aby se našlo co nejvíce chyb byl zdokonalen kód, tato verze byla pojmenována jako PHP verze 2. Tato verze měla základní funkčnost jako dnešní PHP (proměnné ve stylu Perlu, zpracování formulářů, možnost vložit HTML kód).

Vývojáři Andi Gutmans a Zeev Suraski v roce 1997 přepsali parser a vytvořili základ PHP 3. Poté začali opět přepisovat jádro PHP a vydali Zend Engine v roce 1999. 22. Května 2000 byla vydána verze PHP 4, která stojí na novém Zend Engine 1.0. 13. června 2004 byla představena verze PHP 5, stojící na Zend Engine II. PHP 4 se již nevyvíjí a nevydávají se žádné bezpečnostní aktualizace. Od roku 2008 je PHP 5 jedinou stabilní verzí, která se vyvíjí.

4.2 Výhody PHP

Kód PHP může snadno prolínat se statickým obsahem zbytku dokumentu. Lze tak snadno vytvořit stránky obsahující jak dynamický kód, tak statický XHTML. Jeho zdrojový kód lze kdykoli upravovat bez nutnosti vlastnit speciální vývojové nástroje. Vyniká také svou rychlostí v porovnání s konkurenčními technologiemi. Je snadno dostupné, je nabízeno na velkém množství serverů pro webhosting. PHP má schopnost připojit se k mnoha různým databázím. Obsahuje knihovny pro práci s mnoha různými účely. Může pracovat s obrázky, PDF dokumenty, síťovými protokoly atd.

4.3 Nevýhody PHP

Jelikož se jedná o nekompilovaný jazyk, může dojít k tomu, že kdokoli s přístupem k serveru, kde jsou umístěny skripty, může zobrazit jejich obsah. Také rychlost v porovnání s kompilovanými skripty rozhraní CGI je nižší.

Pojmenování metod v PHP není konzistentní. Jsou metody, které mají odděleny víceslovné názvy spodním podtržítkem, ale jsou i metody, kdy jsou víceslovné názvy psány v celku. V případě použití editoru bez syntaxe je tak nutná konzultace v manuálu s popisem [10]

jednotlivých metod. Tato nevýhoda je zvýšena velkým množstvím těchto metod, často je dostupno mnoho metod pro vykonání podobných akcí. Např. některé metody využívané při výstupu: print, princ, sprintf, fprintf, vprintf, vsprintf. [10]

5 Jazyk HTML

HTML HyperText Markup Language je textový značkovací jazyk, kterým se tvoří webové stránky. Stránky jsou jednoduché textové soubory, které obsahují text a několik HTML značek, určující význam jednotlivých částí stránky. Obsahuje pokyny, kterými se určuje, jak má být text či obrázek zobrazen na stránce. Tyto pokyny udávají vzhled celého dokumentu.

Dokument může mimo značkování obsahovat další prvky:

- Direktivy – začínající znaky <!, jsou určeny pro zpracovatele dokumentu (prohlížeč).
- Komentáře – pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje).
- Kód skriptovacích jazyků.
- Definice událostí a kód pro jejich obsluhu.

5.1 Struktura dokumentu

Dokument v jazyku HTML má předepsanou strukturu:

- Deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE.
- Kořenový element – element html (značky <html> a </html>) reprezentuje celý dokument.
- Hlavička elementu – obsahuje metadata, která se vztahují k celému dokumentu. Definují například název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení. Hlavička je uzavřena mezi značky <head> a </head>.
- Tělo dokumentu – obsahuje vlastní text dokumentu. Vymezuje se značkami </body> a </body>.

Příklad zdrojového kódu v HTML 4.01:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <!-- toto je komentář -->
  <head>
    <title>Titulek stránky</title> [6]
```

```
</head>
<!-- tělo dokumentu -->
<body>
  <h1>Nadpis stránky</h1>
  <p>Toto je tělo dokumentu</p>
</body>
</html>
```

5.2 Druhy značek

Značky lze z hlediska významu rozdělit na tři základní skupiny:

Strukturální značky – rozvrhují strukturu dokumentu, příkladem jsou odstavce (<p>), nadpisy (<h1>, <h2>). Dodávají dokumentu formu.

Popisné (sémantické) značky – popisují povahu obsahu elementu, příkladem jsou (<title>) nebo adresa (<address>).

Stylistické značky – určují vzhled elementu při zobrazení, typickým příkladem je značka pro tučné písmo ().

Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky (< a >). Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. element (prvek) dokumentu. Například je otevírací značka pro zvýraznění textu a Kabelová kniha je element obsahující zvýrazněný text.

Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu. Značky (takzvané tagy) jsou obvykle párové, přičemž koncová značka je shodná se značkou počáteční, jen před názvem je znak lomítka.

Příklad pro označení odstavce: <p>Text odstavce</p>

Některé značky jsou nepárové – nemají žádný obsah a nepoužívají koncovou značku.

Příklad pro vykreslení vodorovné čáry: <hr>

Tagy mohou obsahovat atributy, které popisují jejich vlastnosti nebo nesou jinou informaci. Příkladem může být odkaz (tag a), jehož atribut “href” říká, kam se uživatel po kliknutí na něj dostane (v tomto příkladu na stránku VŠB-TUO):

```
<a href="http://www.vsb.cz/">text odkazu</a>. [6]
```

6 Open Source

Termín open source (otevřený zdroj) vznikl v roce 1998 poté, co se firma Netscape rozhodla zveřejnit zdrojový kód svého oblíbeného prohlížeče Navigátor. Toto oznámení podnítilo malou skupinu softwarových vývojářů, kteří byli dlouhodobými členy skupiny, jež se měla brzy přerodit v hnutí open source, k tomu, aby formálně oznámili vznik iniciativ OSI (Open Source Initiaves) a definice OSD (Open Source Definition).

Přestože byla ideologie OSI původně podporována pouze komunitou hackerů, začali po uvolnění zdrojového kódu prohlížeče Netscape Navigator přispívat svými nápady a opravami programátoři všech vrstev a postavení. Poselství OSI bylo vysláno a začalo žít vlastním životem. Hlavní proud počítačového světa s ním musel začít počítat.

První operační systém, který lze považovat za operační systém s otevřeným zdrojem, se stal Linux (systém BSD distribuovaný v Berkeley v roce 1989 byl těsně druhý). Za těmito příklady brzy následovaly další programy. Velké softwarové firmy, jako je například Corel, začaly nakonec nabízet rovněž verze svých programů určené pro systém Linux.

Přestože existuje mnoho různých druhů licencí OSI, může jakýkoli software nést pečeť certifikátu OSI, pokud prošel testem OSD (definice otevřeného softwaru). Zmiňované programy jsou dostupné na mnoha webových serverech. Nejoblíbenějším z webů je v tomto ohledu www.sourceforge.net, jenž hostí více než 83 000 projektů otevřeného softwaru.

Programy typu open source jsou dobré, protože:

- Jsou zadarmo. Nejskvělejší na otevřeném softwaru je skutečnost, že jsou volně dostupné široké veřejnosti. Softwaroví vývojáři a programátoři věnují dobrovolně svůj čas zdokonalování existujícího softwaru a tvorbě nových programů. Otevřený software nevyžaduje už ze samotné své podstaty žádné formy licencování nebo prodejních poplatků.
- Jsou nezávislé na platformě a jsou „nezávislé na technologii“. Požadavkem, že otevřený software musí být nezávislý na platformě, zajistila komunita kolem otevřeného softwaru, že takto definované programy může používat prakticky každý. V souladu s definicí OSD (Open Source Definition) předepsané iniciativou OSI (Open Source Initiative) na stránce <http://opensource.org/docs/definition.php> [1]

musí být program typu open source nezávislý na jakékoli „individuální technologii“ nebo na jakémkoli „stylu rozhraní“. Musí být navíc „technologicky neutrální“.

Lze-li software spouštět ve více operačních systémech, jsou v podstatě tato kritéria splněna.

- Nesmí omezovat jiný software. To v podstatě znamená, že pokud je program typu open source šířen společně s dalšími programy, je lhostejné, zda jsou tyto programy typu open source, nebo zda jsou komerčního charakteru. Díky tomu mají softwaroví vývojáři maximální kontrolu nad svou prací, ale především jsou maximálně flexibilní.
- Zahrnují rozmanitost. Rozmanitost myšlenek a kultur jednoduše produkuje lepší výsledky. Z toho důvodu nemohou otevřené programy diskriminovat žádnou osobu nebo skupinu osob, ani žádnou „oblast působnosti“. Například program navržený pro lékaře nesmí být omezen jen pro tuto profesní skupinu a každý musí mít možnost tento program použít a upravit si ho pro vlastní profesní obor.

Kompletní seznam kritérií, jež musí software splňovat, aby mohl být označen jako „otevřený software“, případně více informací o iniciativě OSI či komunitě otevřeného softwaru najdete na webu OSI na adrese www.opensource.org. [1]

7 Linux

Linux je jádrem několika počítačových operačních systémů. Je příkladem svobodného softwaru a vývoje open source softwaru, celý jeho zdrojový kód je volně k dispozici pro veřejnost a kdokoli jej může volně používat, upravovat, distribuovat pod podmínkou, že dodá i zdrojové kódy.

Termín Linux znamenající linuxové jádro (také označováno jako GNU/ Linux), je často používán pro označení celých unixových operačních systémů, které se skládají z linuxového jádra a zároveň z knihoven a nástrojů z projektu GNU a dalších zdrojů. Dalo by se říci, že spojuje základní systém s velkým množstvím aplikačního softwaru.

Základním programovacím jazykem v Linuxu je jazyk C a sada GCC, obsahující překladače pro několik jazyků (C, C++). Součástí programovacích nástrojů jsou i GNU binutils, obsahující nástroje pro překlad jazyka symbolických adres a linkování binárních objektových souborů do spustitelné podoby. Linux se stal velice oblíbenou platformou pro provoz [5]

WWW serverů, velké množství uživatelských aplikací, které se provozují pod tímto systémem, jsou webové aplikace napsané v jazyce PHP.

7.1 Vývoj Linuxu

V 70. letech byly počítače těsně svázané s operačním systémem, kterým byl ve velké většině Unix a jeho varianty. V té době Richard Stallman pracoval na textovém editoru Emacs a byl portován na Unix. Stallman řešil problém přístupu k softwaru. S nápadem svobodného softwaru přišel počátkem 90. let. Šlo o umožnění ostatním sdílet software. Unix se skládá z jádra (kernelu) a dalších obslužných programů, dohromady tvořící celek. Pokud by jádro a programy přepsal, tak by vznikl nový Unix. Tuto myšlenku pojmenoval GNU. V roce 1983 veřejně vyzval programátory, aby se zapojili do projektu GNU s cílem přepsat Unix a jeho programy. V roce 1985 byl projekt oficiálně představen veřejnosti, vysvětlil jeho principy a profit pro jednotlivce i firmy. Tehdy také zmínil pojem free software, což znamená svobodný software, který může být libovolně šířen. V roce 1988 Stallman získal licenci, tzv. GNU General Public Licence (GPL). Hlavní myšlenka je tzv. copyleft, tzn. Zákaz přivlastnit díla a jejich dalšího rozmnožování bez poskytnutí zdrojových kódů a licence. Zůstává však zachováno autorské právo, autor může dílo prodávat, vydat jinak či ukončit jeho vývoj. Nezakazuje dílo prodávat, kupujícímu musí být poskytnut zdrojový kód tak, aby šel přepsat, modifikovat.

V roce 1991 Linus Torvalds pracoval na jednoduché implementaci unixového systému Minix. V září publikoval svou první verzi Linuxu (0.0.1) na internetu. Linux byl vydán pod licencí GPL a byl plně funkční.

Od této doby vzniklo již několik verzí tohoto systému, dnes je tento systém jeden z nejoblíbenějších.

Výhody

- Linux a mnoho aplikací zdarma,
- Bezpečný systém,
- Otevřené zdrojové kódy,
- Velmi stabilní systém, špičkové uživatelské grafické rozhraní,
- Souborový systém, který se nemusí nefragmentovat,
- Velká škála nastavitelnosti. [5]

Nevýhody

- Méně software,
- Slabší podpora češtiny,
- Absence některých ovladačů,
- Nekompatibilita s programy Windows

7.1.1 Distribuce

Distribuce je soubor programů a nástrojů dodávaný spolu s Linuxem. Dají se rozdělit na dva základní typy:

binární – instalují se, jsou mezi nimi i komerční,

zdrojové – neinstalují se, ale musí se zkompilovat.

Distribuce se mezi sebou liší ve skladbě programů, způsobu a frekvenci vydávání aktualizací, v instalačním programu, konfiguračních nástrojích, v řešení startovacích skriptů, v úpravě některých programů a jádra.

7.1.1.1 Binární

Debian GNU/Linux - tato distribuce je nekomerční, je rozšířená. Nemusí být jen Debian Linux, ale také Debian Hurd a Debian FreeBSd.

Tato distribuce se dá rozdělit na tři vývojové jádra: Stable – nejsilnější a nejstarší software, Unstable – nejnovější a nestabilní a Testing.

Red Hat GNU/Linux - nejznámější distribuce na světě, nyní tato verze je placená.

Fedora GNU/ Linux - je svobodná distribuce, klade důraz na bezpečnost a otevřenost.

Mandriva GNU/Linux (dříve Mandrake) - první verze vznikla v roce 1998, dnes je nejrozšířenější distribucí na světě, je vhodná pro začínající uživatele. [5]

Ubuntu GNU/Linux - distribuce Ubuntu byla vytvořena v roce 2004 společností Canonical, jejímž majitelem a zakladatelem je jihoafrický miliardář Mark Shuttleworth. Je založena na distribuci Debian GNU/Linux, ale liší se od něj zaměřením na uživatelskou přívětivost, snazším použitím a pravidelným vydáváním. Nové verze Ubuntu vychází každý půlrok, zpravidla v dubnu a v říjnu – vždy měsíc po vydání nové verze grafického prostředí GNOME.

Jednou za dva roky – zatím pouze v červnu 2006 a dubnu 2008 – je vytvořeno vydání s dlouhodobou podporou (LTS – Long Term Support), které je určeno především pro [3]

servery, firmy a obecně uživatele, kteří mají zájem provozovat toto konkrétní vydání po velmi dlouhou dobu. Tato LTS vydání jsou podporována tři roky na běžných počítačích a pět let na serverech (oproti 18 měsícům u běžných vydání).

Každé vydání Ubuntu dostane přezdívku v podobě názvu nějakého zvířete s přívlaskem, jež začínají na stejné písmeno – postupuje se podle abecedy. Například současné vydání se jmenuje Karmian Koala, což by se dalo do češtiny přeložit jako karmická koala.

Minulé vydání se jmenovalo Jaunty Jackalope (veselý zajdačen). Zároveň je každé verzi přiděleno číslo, a to podle měsíce vydání v říjnu 2009 nese číslo 9.10 (nikoliv 9.1!) a dubnový Jaunty Jackalope byl 9.04.

Ubuntu se od Debianu a dalších plně komunitních distribucí odlišuje také tím, že za ním stojí komerční firma – Canonical. Canonical vydává Ubuntu zdarma a má v úmyslu vydělávat na placené podpoře systému. Canonical ve svém slibu zaručuje, že Ubuntu od něj bude vždy dostupné zdarma a nebude nabízet žádné speciální verze pro platící zákazníky.

Odvozeniny a varianty Ubuntu

Kromě základního Ubuntu, existuje i několik dalších verzí, takzvaných derivátů. Některé vytváří přímo společnost Canonical a vývojáři Ubuntu, jiné připravují různí lidé z celého světa. V krátkosti představím jen ty nejznámější a nejpoužívanější. Nejčastěji se od klasického Ubuntu liší svým pracovním prostředím (tedy vzhledem a ovládáním) a [3] výběrem softwaru. Někdy je hlavním rozdílem „pouhé“ zaměření na jiné uživatele nebo počítače. I když je odlišnost těchto derivátů často zdánlivě obrovská, pořád se v podstatě jedná o jeden a ten samý systém – nepříliš hluboko pod tím vším, co se zdá být rozdílné, se nachází stejné jádro a stejné ovládače. A proto byť jen mírně pokročilého uživatele není problém udělat v několika okamžicích z jedné varianty jinou.

Kubuntu - nejznámějším a také nejstarším derivátem Ubuntu je Kubuntu, které se od základu liší především grafickým prostředím. Zatímco Ubuntu používá GNOME, Kubuntu upřednostňuje prostředí KDE. Zároveň jsou v souvislosti se změnou grafického prostředí nahrazeny některé výchozí programy. Stejně jako Ubuntu je i Kubuntu vydáváno přímo společností Canonical (s podporou nemalé aktivní komunity), která platí i několik jeho vývojářů. Více informací o Kubuntu naleznete na stránkách <http://www.kubuntu.cz>.

Xubuntu - je zaměřeno především na méně výkonné a starší počítače, protože není tak náročné na systémové prostředky. Namísto prostředí GNOME, které je poněkud náročnější na výkon počítače, totiž používá odlehčený (ale jinak zcela plnohodnotný) grafický systém [3]

Xfce. Opět jsou zde nahrazeny některé výchozí aplikace podle toho, které v Xfce pracují lépe. Více informací o Xubuntu naleznete na stránkách <http://www.xubuntu.cz>.

Edubuntu - neboli Ubuntu Education používá stejně jako Ubuntu prostředí GNOME, ale je zaměřeno na vzdělávání, určené domů i do škol (odtud také pochází název). Obsahuje navíc softwarové balíčky zaměřené na výuku a školství. Původně se jednalo o zcela samostatnou distribuci s vlastním CD, ale od této varianty se postupně upustilo a nyní je Edubuntu dostupné pouze v podobě tzv. přídatného CD, což je specializované CD určené k přeměně klasického Ubuntu na Edubuntu. Více informací a také přídatné CD naleznete na stránkách <http://www.edubuntu.cz>.

Ubuntu Studio - se, jak už jeho název napovídá, zaměřuje na umělce, konkrétně pak na zpracování hudby, videa a grafiky. Tomu je přizpůsobeno grafické rozhraní a především výběr výchozích aplikací. Popularita a kvalita této speciální verze Ubuntu je taková, že ačkoli začínala jako nezávislý projekt, postupně se stala oficiální součástí projektu Ubuntu. Více informací o Ubuntu Studio můžete nalézt na stránkách <http://ubuntustudio.org/>. [3]

Slackware GNU/Linux - patří mezi nejstarší distribuce.

OpenSUSE GNU/Linux - tato verze je placená, je stabilní a také vhodná pro začátečníky. Je to obdoba Fedory.

Arch Linux - distribuce pro pokročilejší uživatele, je velmi rychlá. Pravidelně se aktualizuje z internetu na nejnovější verze programů.

Linspire GNU/Linux - je možné ji nahrát na více počítačů.

7.1.1.2 Zdrojové

Gentoo GNU/Linux - tato distribuce je založena na zdrojových kódech, každý si přeloží svůj systém podle svých požadavků. Obsahuje systém balíčků zvaný Portage, který řeší závislosti a předání parametrů pro překlad.

7.1.1.3 Live

Slax GNU/Linux - tato distribuce se dá spustit bez instalace, hodí se spíše na ukázkou systému nebo na opravu nainstalovaného systému. [5]

8 Apache

Webový multiplatformní server Apache je v současné době nejvíce používaným webovým serverem, a to na více než polovině instalací webových serverů dodávajícím prohlížečům internetové stránky. Je šířen zdarma i se zdrojovými kódy. Je využíván pro GNU/Linux, BSD, Solaris, Microsoft Windows a další.

Důležité moduly serveru Apache jsou např.:

- Mod_SSL – šifrovaná komunikace mezi serverem a klientem
- Mod_PHP – modul umožňující použití skriptovacího jazyka PHP
- Mod_Python – modul skriptovacího jazyka Python
- Mod_Pperl – modul skriptovacího jazyka Perl

Vývoj Apache začal v roce 1993 na Illinoiské univerzitě. Původní název projektu byl NCSA HTTPd. Hlavní úlohu ve vývoji sehráli Brian Behlendorf a Cliff Skolnick, kteří založili e-mailovou konferenci a začal sběr úprav. První veřejná verze s označením 0.6.2. byla vydána v dubnu 1995. Následovalo kompletní přepsání kódu a založení Apache Group, která je dnes základem vývojářského týmu. Od dubna 1996 byl Apache nejpoblárnější server na internetu.

V květnu 1999 běžel na 57% všech serverů a v listopadu 2005 jeho používanost byla 69%. (Výsledky měření Netcraft). V současné době dodává prohlížečům na celém světě většinu internetových stránek. [2]

9 PhpMyAdmin

PhpMyAdmin je webová aplikace napsaná v PHP a obsahující klientský kód XHTML, CSS a JavaScriptu. Poskytuje kompletní webové rozhraní pro administraci databází MySQL a je ve svém oboru obecně považován za vůdčí aplikaci.

Protože je od svého vzniku open source, těší se podpoře četných vývojářů a překladatelů po celém světě. Hostitelem projektu je v současné době SourceForge.

Produktu phpMyAdmin (můžeme ho najít na adrese <http://phpmyadmin.cz/>) prokázali důvěru hostitelští poskytovatelé po celém světě tím, že ho naistalovali na své servery. Kromě toho můžeme instalovat svou vlastní kopii phpMyAdmin dovnitř našeho webového prostoru, [4]

pokud má náš poskytovatel nainstalované nejnovější verze PHP a MySQL. Populární CPanel (řídící aplikace webu) je s phpMyAdmin v souladu.

9.1 Souhrn schopností phpMyAdmin

Cílem phpMyAdmin je nabídnout kompletní správu serverů a dat MySQL založenou na webech, a držet krok s evolučními standardy MySQL a webu. Přestože produkt není perfektní, aktuálně zahrnuje nejběžnější požadované schopnosti a spoustu věcí navíc.

Vývojový tým trvale produkty vyvíjí na základě oznámených chyb a požadavků na schopnosti a pokračuje ve vydávání podverzí v intervalu zhruba dvou měsíců.

PhpMyAdmin nabízí schopnosti, které pokrývají základní operace MySQL s databázemi a s tabulkami. Má také interní relační systém, který udržuje metadata za účelem podpory vyspělejších schopností. Konečně, systémoví administrátoři mohou z phpMyAdmin spravovat uživatele a oprávnění. Je důležité připomenout, že volby dostupných operací phpMyAdmin závisejí na oprávněních, která byla uživateli udělena ke specifickému serveru MySQL.

Základní výbavu tvoří následující schopnosti:

- Vytvoření a odstranění databáze
- Vytvoření tabulky, přejmenování, kopírování a odstranění
- Údržba struktury tabulky, včetně indexů
- Speciální operace nad tabulkou (oprava, optimalizace, změna typu)
- Vkládání dat, modifikace, odstraňování
- Zobrazování dat vodorovně, svisle a náhled před tiskem
- Navigace po datech a třídění
- Nahrávání binárních dat
- Vyhledávání dat (v tabulce nebo databázi)
- Dotaz podle příkladu (nad několika tabulkami)
- Dávkové načítání dat
- Export struktury a dat v rozličných formátech, s komprimací
- Instalace pro více uživatelů a více serverů. [4]

Do nadstavby patří:

- Komentáře na úrovni sloupců
- Cizí klíče (s INNODB nebo bez)
- Prohlížení připojené tabulky
- Oblíbené dotazy
- Slovník dat
- Ralační schéma a slovník PDF
- Historie dotazů SQL
- Připojení k MySQL buď pomocí tradičního rozšíření „mysql“, nebo nového rozšíření mysqli (vPHP5)
- Podpora znakových sad pro databáze, tabulky a sloupce (s MySQL 4.1)
- Transformace obsahu sloupce založené na typu MIME
- Správa vzhledů, jimiž se dá přizpůsobit vzhled rozhraní.

Do schopností pro administraci serveru patří:

- Správa uživatelů a oprávnění
- Kontrola databázových oprávnění
- Stav serveru
- Export více databází najednou. [4]

10 Virtuální počítač

Virtuální počítač – virtuální stroj je vlastně software, který vytváří vizualizované prostředí mezi platformou počítače a operačním systémem, kde může uživatel provozovat software na abstraktním stroji.

Dalo by se říci, že pomocí virtualizace se dá jeden stroj (server, procesor, paměť, síťová karta) využít pro více než jeden operační systém.

10.1 Hardwarový virtuální stroj

Hardwarový virtuální stroj označuje několik totožných pracovních prostředí na jednom počítači, z nichž na každém běží operační systém. Aplikace psaná pro jeden operační systém může být používána na stroji, kde běží jiný operační systém. Výhodou je to, že start a restart virtuálního počítače může být rychlejší, protože mohou být přeskočeny úkoly jako např. inicializace hardware.

10.2 Aplikační virtuální stroj

Jedná se o počítačový software, který izoluje aplikace používané uživatelem na počítači. Verze virtuálního stroje jsou psány pro různé platformy, jakákoli verze psaná pro virtuální stroj může být provozována na jakékoli z platform, místo toho, aby se musely vytvářet oddělené verze aplikace pro každý počítač a operační systém.

10.3 Virtuální prostředí

Virtuální prostředí (virtuální soukromý server) je jiný druh virtuálního stroje. Je to vizualizované prostředí pro běh programů na úrovni uživatele (tj. aplikace). Virtuální prostředí jsou vytvořena použitím softwaru zavádějícího vizualizaci na úrovni operačního systému, např. Virtuozzo, FreeBSD Jails, Linux-V Server.

11 VMware

VMware (Virtual Machine) je produkt firmy VMware Inc., sloužící pro virtualizaci jednoho nebo i více počítačů na jednom hostitelském stroji.

VMware nazývá počítač, který vykonává veškeré operace jako „hostitelský“ a operační systémy dělicí prostředky hostícího počítače jako hosty.

Jako emulátor (druh software umožňující běh počítačových programů na jiné platformě než pro kterou byly původně vytvořeny a kterou samy od sebe podporují) poskytuje hostům kompletní sadu virtualizovaného hardwaru – procesor, paměť, grafické adaptéry, síťové karty, ovladače disků. Program je schopen vytvořit virtuální procesor, nahradit část paměti, disk a použít některé periferie hostitelského systému tak, aby si spouštěné systémy (programy) „myslely“, že byly spuštěny na standardním počítači platformy PC.

VMware podporuje virtualizaci Windows, Linuxu, Solarisu, FreeBSD.

Virtuální nástroje VMware jsou snadno přenositelné mezi různými hostitelskými počítači. Může se zastavit operace prováděna virtuálním strojem, zkopírovat ji na jiný hostitelský počítač a znovu v ní pokračovat. [11]

12 Realizace

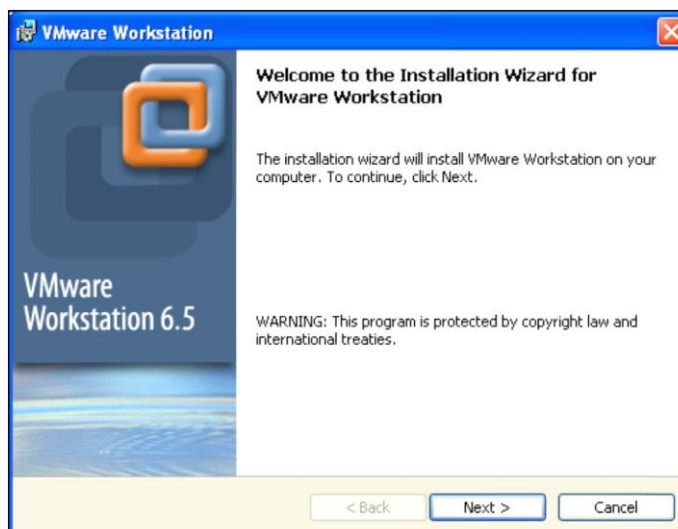
Mým úkolem je vytvoření webové aplikace pro správu kabelů v elektronické podobě. Nejdříve je třeba určit software, se kterým budu pracovat. Na základě zadání, má aplikace pracovat na platformě Linux, ale na mém počítači mám nainstalován operační systém Windows, proto bylo třeba najít řešení, abych nemusel přinstalovat celý počítač na Linux. Možnost, jak tento problém vyřešit, je virtuální PC, které využívá hardware původního počítače a vytváří zcela samostatný počítač s vybraným operačním systémem podle zadání. Můžeme vybrat z několika možností od různých společností, ale po předchozích zkušenostech jsem zvolil software VMware a její produkt VMware workstation.

12.1 Instalace virtuálního počítače

Instalace je jednoduchá a zvládne ji každý. Nejdříve si seženeme instalační soubor. Je možné ho stáhnout například zde:

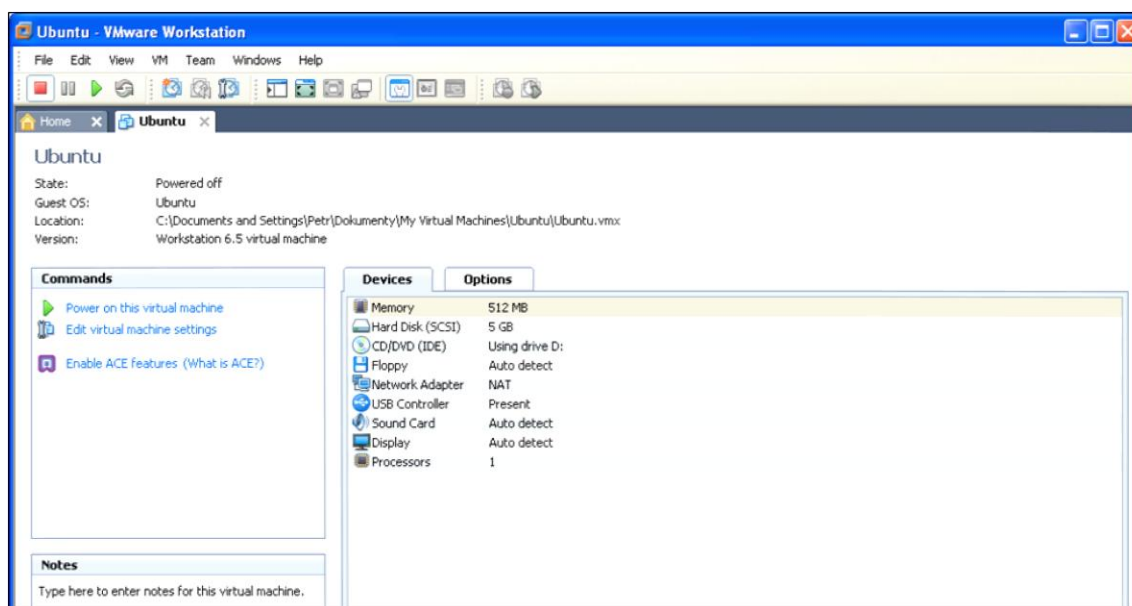
http://downloads.vmware.com/d/details/wkst_654_win/dHdiZColcGJkZEAq

Tento software není open source, ale je možné ho nainstalovat bez nutnosti registrace nebo zadání výrobního klíče a na 30dní si ho vyzkoušet. Pak se můžeme rozhodnout, zda uvedený software budeme používat nebo si vybereme jiný. Pro odzkoušení mé aplikace, stačila 30-ti denní zkušební doba. Můžeme přejít k instalaci virtuálního PC a spustíme instalační soubor, vše proběhne v šesti krocích. Krok 1 – spuštění instalace, zobrazuje obrázek 1. V kroku 2 - výběr druhu instalace, zvolil jsem typickou. Krok 3 – výběr umístění instalace, vybereme



Obr. 1. Spuštění instalace VMware workstation

místo, na našem počítači, kde se má instalovat software. Krok 4 – výběr ikon, které se mohou nainstalovat, např. ikona pro spouštění softwaru na pracovní ploše. Krok 5 – potvrzení spuštění instalace, ještě je možné se vrátit a udělat potřebné změny, například jiné umístění. Proveďte se instalace softwaru na náš počítač a posledním krokem, je krok 6 – ukončení instalace. Poté program vyžaduje provedení restartu počítače. Software je připraven k použití. Spustíme virtuální počítač a na úvodní stránce si můžeme přečíst, jaký hardware používá. Vidíme, kolik paměti RAM, velikost HDD, jaké mechaniky a další periferie počítače, které používá viz. obrázek 2.



Obr. 2. Spuštěný virtuální počítač VMware workstation

12.2 Instalace operačního systému

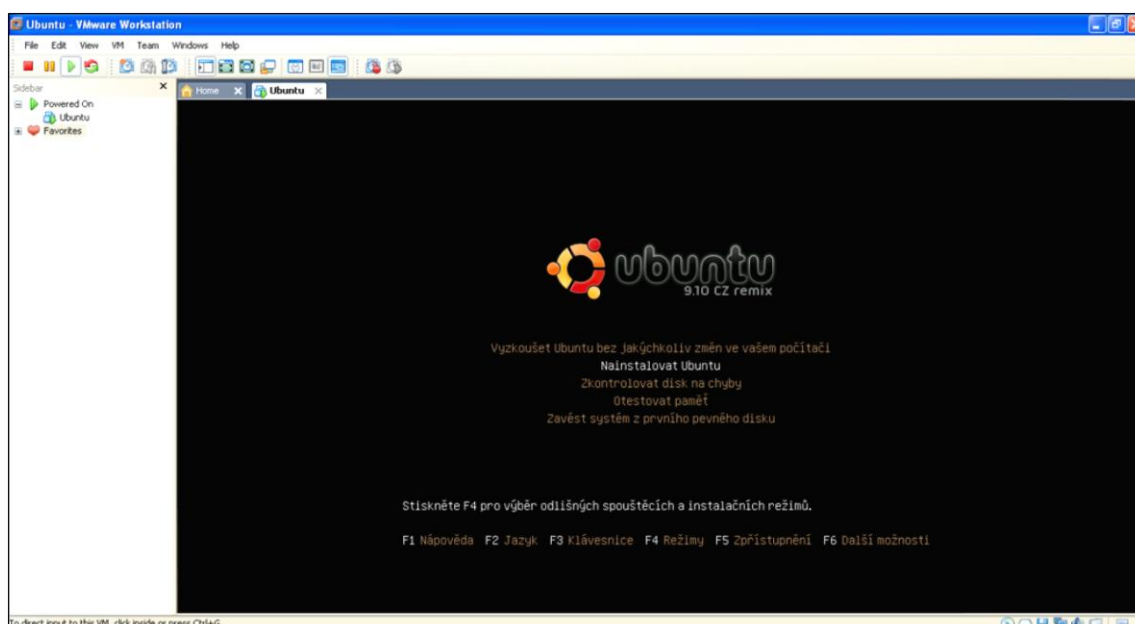
Na náš nový virtuální počítač musíme nainstalovat operační systém, stejně jako na nový počítač. Po svých dobrých zkušenostech jsem si vybral distribuci Linux Ubuntu, která je celá v české lokalizaci a technickou podporu má také češtině. Další z předností, tento software je open source, takže si ho můžeme stáhnout zdarma na adrese:

<http://www.ubuntu.cz>

Zakoupil jsem si uživatelskou příručku, která obsahovala instalační DVD, což se mi zdá, jako nejlepší řešení. Můžeme přistoupit k instalaci operačního systému a popsat její kroky. Spustíme ji tlačítkem „New Virtual Machine“, objeví se okno VMware. Krok 1 – výběr typu instalace, vybral jsem typickou. Krok 2 - vložení instalačního DVD do mechaniky. Krok 3 – načtení instalačního DVD, zjistí co je to za DVD a můžeme pokračovat dalším krokem.

Krok 4 – pojmenování našeho nového virtuálního počítače a výběr umístění složky VMware.

Krok 5 – nastavení velikosti volného místa, kterou může VMware zabrat z našeho pevného disku, doporučena je 10GB, já jsem zvolil 5GB. V dnešní době, kdy jsou dostupné pevné disky 1TB, se nemusíme omezovat a můžeme zvolit i více, podle toho co ještě budeme do virtuálního počítače instalovat. Po nainstalování operačního systému, provedení všech aktualizací a doinstalování doplňkových balíčků, zabírá distribuce na našem pevném disku 4,7GB volného místa. Krok 6 – kontrola nastavení, ještě je možnost se naposledy vrátit a zadané hodnoty změnit. Krok 7 – výběr co chceme udělat, nyní začíná načítání instalačního DVD a už vše probíhá jako klasická instalace do čistého počítače. Vybral jsem „Nainstalovat Ubuntu“ viz. obrázek 3. Při práci s virtuálním počítačem může dojít k situaci, že se nám ztratí



Obr. 3. Instalace operačního systému Ubuntu do VMware workstation

kurzor myši. Je třeba dávat pozor, kde se kurzor nachází. V prostoru pracovní plochy virtuálního počítače, používáme jeho prostředky, a když přejdeme na okraj okna nebo menu VMware workstation, tak pracujeme na skutečném počítači. Tento stav není nikde signalizován, jen drobnou změnou tvaru kurzoru, kterou používá distribuce Ubuntu. Jinak přechod mezi virtuálním počítačem a vlastním počítačem je okamžitý. Pokud se kurzor ztratí, je nutno ho resetovat stiskem kláves „Ctrl + Alt“. Následuje další krok, krok 8 – výběr jazyka instalace, zvolený jazyk bude zároveň i výchozím jazykem systému instalace. Krok 9 – výběr časového pásma, zvolená oblast „Evropa“ a pásmo „Česká republika“. Krok 10 – výběr rozložení klávesnice, záleží na každém, jakou klávesnicí si zvolí, která mu vyhovuje.

V podstatě se jedná jenom o prohození kláves „Z“ a „Y“, zadanou klávesnici si můžeme ihned odzkoušet v textovém poli. Krok 11 – výběr umístění instalovaného systému, není třeba nic měnit, instalace se provede do prostoru, který jsme zvolili pro VMware workstation. Krok 12 – pojmenování virtuálního počítače a zvolení hesla, je třeba upozornit, že heslo je důležité a bude potřeba pro další instalaci balíčků do operačního systému, takže je dobré, si ho někam zapsat. Tento krok vidíme na obrázku 4. Krok 13 – rekapitulace, poslední možnost něco upravit, před spuštěním instalace operačního systému do virtuálního počítače. Krok 14 – instalace operačního systému, nyní se spustí instalace operačního systému, která trvá okolo 30 minut. Podobně jako u běžného počítače. Dalším krokem je restart virtuálního počítače a je nutné vyjmout instalační DVD z mechaniky. Poté se spustí nový operační systém a můžeme začít s používáním našeho nového virtuálního počítače. Všechno potřebné pro práci s počítačem je nainstalováno, jako je např. internet, ovládače a je připraveno k okamžitému použití. Operační systém si najde aktualizace, od posledního vydání. K potvrzení instalace stažených aktualizací musíme zadat heslo, které jsme vyplňovali při instalaci operačního systému, a poté nás požádá o provedení dalšího resetu virtuálního počítače. Tímto jsme dokončili instalaci operačního systému a můžeme přejít k instalaci balíčku dalšího softwaru.

Kdo jste?

Jaké je vaše jméno?

Jaké jméno chcete použít pro přihlášení?

Pokud bude tento počítač používat více než jedna osoba, můžete nastavit více účtů po instalaci.

Vyberte heslo, abyste udrželi svůj účet v bezpečí.

Vložte stejné heslo dvakrát, aby mohlo být ověřeno na překlepy. Dobré heslo by mělo obsahovat směs písmen, číslic a interpunkce. Mělo by být nejméně 8 znaků dlouhé a měli byste jej pravidelně měnit.

Jaké je jméno tohoto počítače?

Toto jméno bude použito, pokud počítač zpřístupníte ostatním na síti.

☒ Přihlašovat se automaticky
☐ Požadovat mé heslo pro přihlášení
☐ Požadovat mé heslo pro přihlášení a rozšifrování mé domovské složky

Krok 5 z 6

Obr. 4. *Pojmenování virtuálního počítače a zadání hesla*

12.3 Instalace LAMP

Další software, který budeme potřebovat pro práci s databází a PHP skripty, je server, pak správce databáze a PHP. Na výběr, jak již bylo uvedeno v teoretické části, máme z velkého množství softwaru, od různých společností. Výběr byl zaměřen do řad open source

softwaru a na tomto základě jsem zvolil Apache server, MySQL a PHP5. Daný software, jde nainstalovat samostatně nebo dohromady v různých trojkombinacích. Veškerý software je obsažen v LAMP serveru, kde písmeno L označuje, pro jaký systém je server určen L – Linux. Dále obsahuje A – Apache server, M – MySQL a P – PHP, takže vše, co budeme potřebovat. Tento způsob je přístupnější, poněvadž nemusíme dělat větší úpravu softwaru, pro jejich další spolupráci. Navíc se nemusíme starat o to, kde potřebný software stáhneme a jakou verzi, aby vše správně fungovalo.

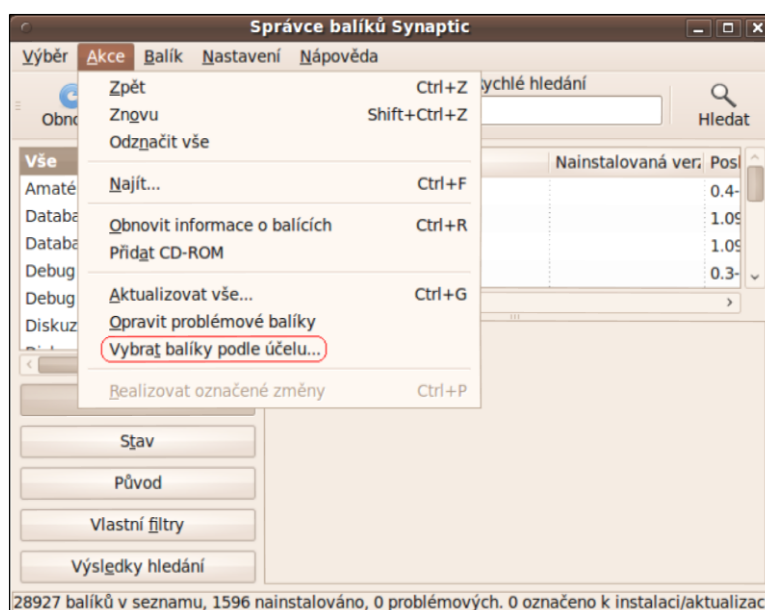
Pro úplnost a toho, kdo je zdatnější, uvádím internetové adresy, kde si může potřebný software stáhnout samostatně.

Apache server je dostupný na adrese: <http://www.apache.org>

MySQL je dostupný na adrese: <http://www.mysql.com>

PHP je dostupné na adrese: <http://www.php.net>

Přejdeme k instalaci LAMP serveru, nejjednodušší způsobem, jak LAMP server nainstalovat, je pomocí správce balíků Synaptic. Správce balíků Synaptic v operačním systému Ubuntu, najdeme zde: **Systém / Správa > Správce balíků Synaptic**. Spustíme program Synaptic, před jeho spuštěním, vyžaduje systém autorizaci heslem, které jsme zadali při instalaci. Nejdříve pomocí tlačítka „**Obnovit**“, provedeme aktualizaci seznamu balíků. Pak musíme najít LAMP server, který najdeme zde: **Akce / Vybrat balíky podle účelu...** a vybereme **LAMP server**. Stejně, jak můžeme vidět na obrázku 5.



Obr. 5. Instalace LAMP serveru pomocí správce balíků Synaptic

Zaškrtnutím LAMP serveru, správce balíků Synaptic, vybere všechny potřebné balíky pro jeho instalaci a tlačítkem „**Vybrat**“, potvrdíme jejich výběr. Tlačítkem „**Použít**“, zahájíme jejich instalaci, ale nejdříve se můžeme podívat, jaké balíky jsou vybrány a popřípadě udělat jejich změnu. Znovu potvrdíme tlačítkem „**Použít**“, program je nejdříve stáhne a následně nainstaluje. Během instalace MySQL, jsme požádáni o vyplnění hesla pro uživatele root. Heslo se používá pro přístup do databáze pomocí programu phpMyAdmin. Dále se používá ve skriptech PHP, kde je nutné se připojit k databázi, za pomoci tohoto hesla. Další velmi důležitá věc pro fungování webové aplikace. Program podruhé požádá o vyplnění hesla, zda je zadané ve správném tvaru, jak jsme požadovali. Abychom dokončili instalaci LAMP serveru, je možné ještě doinstalovat několik doplňkových balíčků, které můžeme potřebovat nebo si vybrat další, podle toho co ještě budeme potřebovat.

12.3.1 Apache moduly

Zde uvádím příklad instalace modulů Apache. Každý program má spoustu doplňkových modulů, které je možné doinstalovat samostatně. Balík vybereme ze seznamu programu Synaptic, protože je jich velké množství, tak je můžeme vyhledat zadáním do vyhledávače. Všechny balíky, jsou pojmenované podle daného programu. Ve výchozím nastavení se instaluje Apache2, tedy balíky modulů začínají například takto:

- libapache2-mod-mono
- libapache2-mod-perl2
- libapache2-mod-python

Po nalezení modulu v seznamu balíků, ho zaškrtneme a program Synaptic nám zobrazí, jaké balíky se budou instalovat. Další postup je shodný, jako při instalaci LAMP serveru. Apache server se spouští automaticky, při startu virtuálního počítače. Pokud potřebujeme Apache zastavit, provedeme to z příkazového řádku terminálu (**Aplikace / Příslušenství > Terminál**) a zadáme příkaz: `sudo /etc/init.d/apache2 stop` (je vyžadována autorizace).

12.3.2 PHP moduly

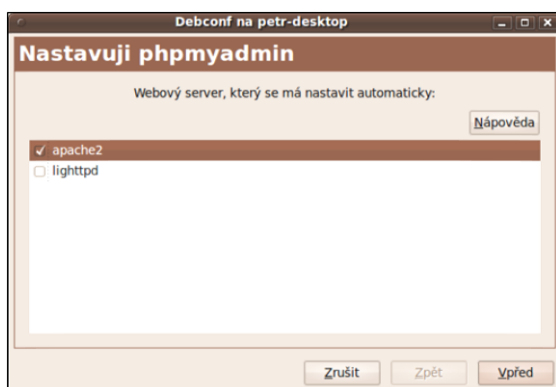
Ve výchozím nastavení LAMP serveru se instaluje PHP5, tedy balíky modulů vždy začínají „php5“, například:

- php5-gd - grafická GD knihovna
- php5-pgsql - podpora PostgreSQL
- php5-sqlite - podpora SQLite
- php5-tidy - knihovna Tidy

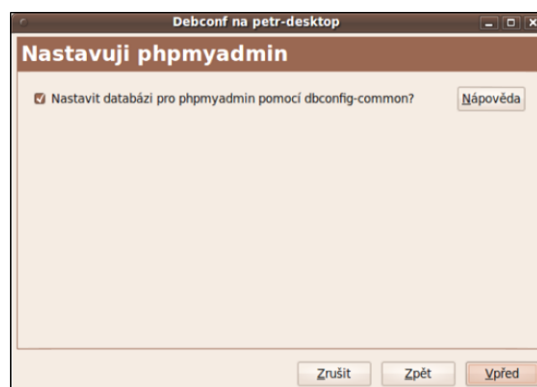
Instalace je stejná, jako u modulů Apache.

12.3.3 PhpMyAdmin

Protože ovládání MySQL přímo pomocí příkazů není příliš pohodlné, existují nástroje, které tuto práci velmi ulehčují. Znova máme na výběr z mnoha různých programů. Například Adminer, MySQL Administrátor a zvolený phpMyAdmin. PhpMyAdmin je zřejmě nejpoužívanější nástroj pro administraci MySQL databáze na webových serverech. Jedná se o webovou aplikaci, takže je možné se na něj dostat i z jiných počítačů. Instalace je stejná, jako u všech předchozích příkladů. Musíme vybrat balík phpmyadmin. Během instalace jsme dotázáni, jaké webové servery se mají s phpMyAdmin nastavit. Vyberte apache2, stejně jako na obrázku 6. a obrázku 6.a..



Obr. 6. Výběr serveru Apache2 pro phpMyAdmin



Obr. 6.a. Výběr nastavení databáze pro phpMyAdmin

Zde končí instalace veškerého potřebného softwaru, pro práci s webovou aplikací.

12.4 Ověření funkčnosti instalovaného softwaru

První co je dobré udělat po instalaci veškerého softwaru, je odzkoušet, zda je všechno v pořádku a nainstalovaný software pracuje správně.

12.4.1 Ověření Apache serveru

Server Apache se spouští automaticky po nainstalování nebo po spuštění systému. Ověření jestli nám pracuje Apache server, je jednoduché. Spustíme webový prohlížeč (v případě Linuxu, Firefox) a jako adresu zadáme: <http://localhost>. Výsledkem by mělo být zobrazení webové stránky s nápisem It works!, stejně jako na obrázku 7.

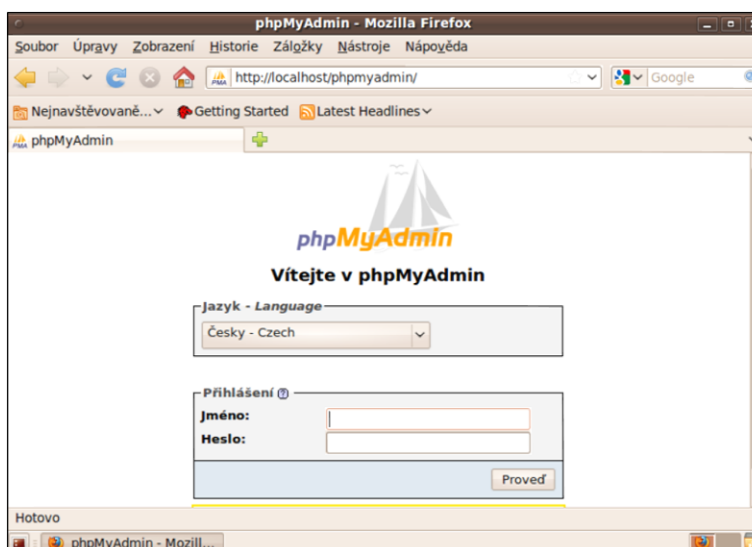


Obr. 7. Ověření, jestli nám pracuje Apache server

Pokud budeme k Apache serveru přistupovat z jiného počítače, tak místo adresy localhost, zadáme IP adresu počítače, na kterém je Apache server nainstalován.

12.4.2 Ověření MySQL

MySQL se také spouští automaticky po instalaci nebo po startu systému. Pro administraci MySQL jsem použil PhpMyAdmin, je to webová aplikace, takže do adresy webového prohlížeče zadáme: <http://localhost/phpmyadmin/>. Na webové stránce se zobrazí formulář pro přihlášení do programu, stejně jako na obrázku 8.



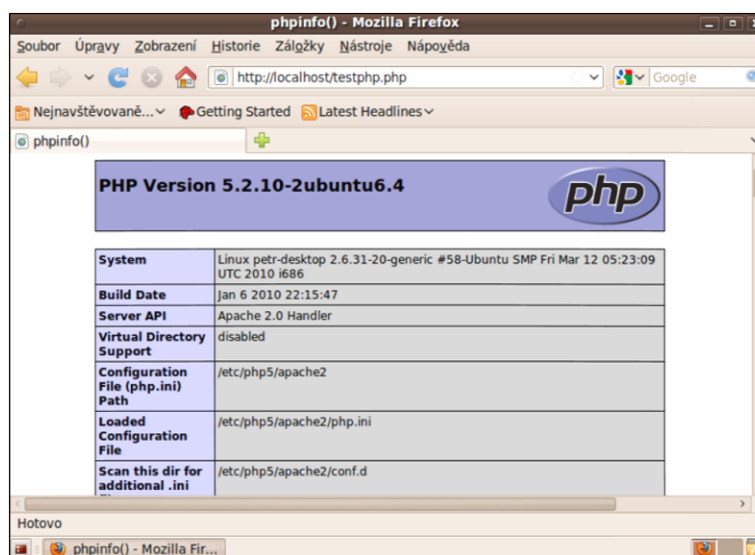
Obr. 8. Ověření, jestli nám pracuje phpMyAdmin

Pokud chceme provést autorizaci, použijeme přihlášení pod uživatelem root a heslem, které jsme zadali při instalaci MySQL.

12.4.3 Ověření PHP

Funkčnost PHP ověříme naším prvním skriptem v PHP. Nejdříve si musíme vybrat, v jakém programu budeme naše skripty psát. Úplně si vystačíme s textovým editorem gedit, který najdeme zde: **Aplikace / Příslušenství > Textový editor gedit**. Textový editor zvýrazňuje syntaxi skriptovacího jazyka PHP, který je pak přehledný a snáze se v něm orientuje. Spustíme textový editor gedit a zapíšeme do něj tento řádek: `<?PHP phpinfo(); ?>`

Toto je náš první skript vytvoření ve skriptovacím jazyku PHP. Nesmíme zapomenout zadat příponu php (např.: info.php). Abychom mohli vytvořený skript spustit, musíme ho přemístit do adresáře /var/www. Ve výchozím nastavení nemá běžný uživatel oprávnění zapisovat do adresáře /var/www, kde jsou umístěny soubory serveru. Musíme změnit oprávnění tohoto adresáře. V terminálu zadáme tento příkaz: `sudo chmod -R 777 /var/www`. Pak je vyžadována autorizace. Nyní už můžeme přemístit náš první skript do souboru /var/www. Do adresy webového prohlížeče zadáme: <http://localhost/info.php> a měla by se zobrazit tabulka, kde jsou všechny informace o PHP, stejně jako na obrázku 9.



Obr. 9. Ověření, jestli funguje PHP

Pokud se nám neobjeví tabulka o PHP, tak můžeme ještě provést aktivaci modulu PHP5. Do terminálu, zadáme tento příkaz: `sudo a2enmod php5`.

Pak musíme restartovat Apache server příkazem: `sudo /etc/init.d/apache2 restart`. Nyní máme vše ověřeno a můžeme se pustit do tvorby webové aplikace.

12.5 Vytvoření webové aplikace databáze

Na začátek jsem si vytvořil funkční model databáze v programu Excel. Tímto jsem získal přehled, jakým způsobem se bude webová aplikace vytvářet a od čeho začít. Nejdříve si musíme ujasnit, jaké data budeme používat a jaké informace chceme ukládat do databáze. Začneme vytvořením tabulek databáze.

12.5.1 Vytvoření databáze a tabulek

Začneme vytvořením databáze „kabelová kniha“. Pro práci s databází používáme SQL příkazy. Spustíme program phpMyAdmin a do příkazového řádku SQL, zadáme příkaz pro vytvoření databáze:

```
CREATE DATABASE `kabelova_kniha`  
  
DEFAULT CHARACTER SET latin2  
  
COLLATE latin2_czech_cs;
```

Pokračujeme vytvořením všech tabulek, které budeme používat v naší databázi.

12.5.1.1 Vytvoření tabulek

Začneme vytvořením tabulky „ustredna“. Tabulka bude obsahovat sloupec „HW_pozice“, kde bude uvedena HW pozice telefonní ústředny. HW pozice telefonní ústředny Siemens, která je provozována na VŠB-TU Ostrava, je v tomto tvaru (např.: 1-10-49-60). Sloupec „linka“, do kterého se ukládá telefonní číslo, používáme čtyřmístné číslování. Dále sloupec „typ“, který slouží pro rozlišení, jaké zařízení se dá zapojit na uvedenou HW pozici (např.: 1 – analogová linka, 2 – digitální linka, 3 – modem, 4 – fax atd.). Jako poslední je sloupec „poznámka“, kde je možné uvést doplňující informace (např.: v poruše, odpojeno, blokováno atd.).

Následuje vytvoření tabulky „ustredna“, pomocí SQL příkazu CREATE:

```
CREATE TABLE `kabelova_kniha`.`ustredna` (`HW_pozice` VARCHAR(12) NOT NULL,  
`linka` INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
`typ` INT(2) NOT NULL, `poznámka` VARCHAR(50) NOT NULL,  
UNIQUE (`HW_pozice`)) ENGINE = InnoDB;
```

Další tabulka „kabely_popis“, slouží k definici kabelu. Tabulka slouží jako přehled všech dostupných kabelů, jejich popis, kam vedou a kolik mají páru. Obsahuje sloupec název, ve kterém je pojmenování kabelu (např.: 1-7-23), tento způsob popisu kabelu, se používá na VŠB-TU Ostrava. Pojmenování a vše okolo, je uvedeno v normě ČSN 34 2300. Předpisy pro vnitřní rozvody a sdělovací vedení nebo ČSN 34 51 23. Norma stanoví základní slovenské a české názvosloví. Druhý sloupec název, popisuje kabel, odkud a kam vede (např.: Budova N > Budova A). Třetí a poslední sloupec, uvádí počet páru na daném kabelu. Jako primární klíč, slouží sloupec kabel.

Vytvoření tabulky „kabely_popis“:

```
CREATE TABLE `kabelova_kniha`.`kabely_popis` (`kabel` VARCHAR(7) NOT NULL,  
`nazev` VARCHAR(50) NOT NULL , `par` INT(3) NOT NULL,  
PRIMARY KEY (`kabel`)) ENGINE = InnoDB;
```

V následující tabulce se budou ukládat data, co se nachází na jednotlivých párech kabelu. Máme zde sloupec „kabel_id“, který je použit jako primární klíč. Následuje sloupec „kabel“, který je cizím klíčem z tabulky kabely popis. Třetí sloupec „par“, definuje pár kabelu, na kterém, je linka nebo jiné zařízení, zapojen. Čtvrtý sloupec „linka“, je cizím klíčem z tabulky „ustredna“. Pátý sloupec „kancelar“, specifikuje umístění, do které kanceláře (např.: N211, N319, N201, atd.), daný pár vede. Poslední sloupec „poznámka“, slouží jako doplňující informace o daném zařízení, které je zapojeno (např.: DECT, IP, VoIP, bezdrát, paralelka, atd.).

Vytvoření tabulky „kabely“:

```
CREATE TABLE `kabelova_kniha`.`kabely` (`kabel_id` INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, `kabel` VARCHAR(7) NOT NULL ,  
`par` INT(3) NOT NULL, `linka` INT(4) NOT NULL, `kancelar` VARCHAR(4) NOT NULL  
, `poznámka` VARCHAR(50) NOT NULL, INDEX (`kabel`, `linka`)) ENGINE = InnoDB;
```

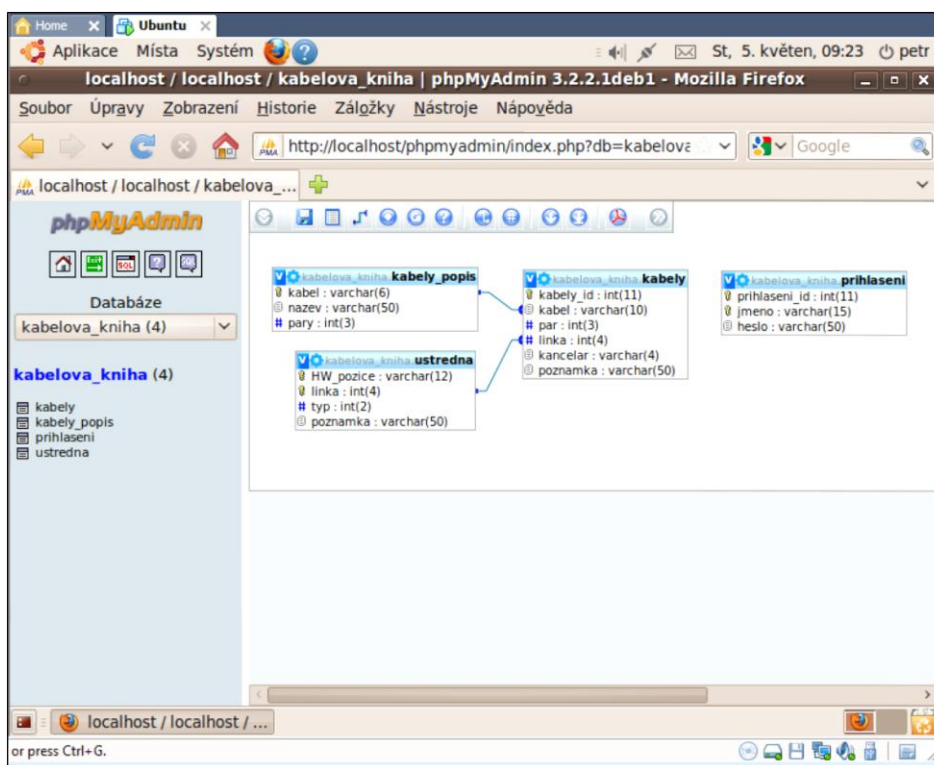
Poslední tabulka nám slouží pro ukládání přihlašovacích hesel uživatelů, které jsou zapotřebí k autorizaci vstupu do kabelové knihy. Tabulka obsahuje sloupec „prihlaseni_id“, které je primárním klíčem. Sloupec „jmeno“, kde je uloženo přihlašovací jméno uživatele. V tomto sloupci jsem použil unikátní klíč, abych zamezil zadání stejného jména dvakrát, do tabulky. Sloupec „heslo“, zde se ukládají zahashované hesla uživatelů, pomocí hashe MD5. Je to z důvodu bezpečnosti uložených hesel.

Vytvoření tabulky „prihlaseni“:

```
CREATE TABLE `kabelova_kniha`.`prihlaseni` (`prihlaseni_id` INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, `jmeno` VARCHAR(15) NOT NULL ,  
`heslo` VARCHAR(50) NOT NULL, UNIQUE (`jmeno`)) ENGINE = MyISAM;
```

12.5.1.2 Propojení tabulek

Když máme vytvořené všechny tabulky, tak je třeba tyto tabulky propojit mezi sebou pomocí primárního a cizího klíče. Propojení tabulek jsem provedl za pomoci programu phpMyAdmin v záložce „Návrhář“. Vstupní heslo do programu phpMyAdmin je stejné, jako to, které používám ve skriptech PHP, a to uživatel „root“ a heslo „12345“. Pokud se nepřihlásíme pod tímto heslem, tak nemůžeme tabulky propojit, protože se nám nezobrazí záložka „Návrhář“ a nebudeme přihlášení jako administrátor. Jedná se v podstatě o spojení tří tabulek. V tabulce „ustredna“ máme primární klíč „linka“ a tento primární klíč vložíme jako cizí klíč do tabulky „kabely“. V tabulce „kabely_popis“ máme jako primární klíč zvolený sloupec „kabel“ a ten vložíme do tabulky „kabely“. Propojení všech tabulek v programu phpMyAdmin máme na obrázku 10.



Obr. 10. Propojení tabulek databáze „kabelova_kniha“

Máme vytvořený základ naší databáze. Můžeme přejít k tvorbě skriptu webové aplikace.

12.5.2 Tvorba skriptu webové aplikace

Podle vzoru, který byl vytvořen za pomoci Excelu, jsem získal představu, jak by se dala vytvořit, tato webová aplikace. Základem webové aplikace jsou formuláře, do kterých zadáváme potřebné informace pro uložení dat do tabulek databáze. Záznam je zpracován

pomocí PHP skriptu. PHP provádí všechny operace, které jsou vyžadovány např.: uložení dat do tabulky, opravení dat v tabulce, vypsání uložených dat nebo vyhledávání dat v tabulce. PHP skript může obsahovat tři různé programovací jazyky jako je HTML kód, PHP kód a příkaz SQL. Můžeme si to prohlédnout názorně v příkladu, kde je výpis ze skriptu pro hledání linky:

Stránka začíná HTML kódem

```
<HTML>
<TITLE>Vyhledani telefonni linky</TITLE>
<h2><font color="#008000">Vyhledani telefonniho cisla: <? echo $_POST["hledej"]; ?></font></h2>
...
```

Pokračuje PHP kódem

```
<? do { @ $spojeni = MySQL_Connect("localhost", "root", "12345"); // připojení do databáze
        if (!$spojeni):                                     // nepodařilo se připojit do databáze
            echo "Nepodařilo se připojit k MySQL.<BR>\n";      // tak vypiš tuto zprávu
            break;                                           // zastav skript
        endif;
    } while (true);
    ...
```

Zde provede SQL dotaz

```
@ $vysledek = MySQL_Query("SELECT * FROM ustredna WHERE linka=$hledej"); // proved' výběr
z tabulky „ustredna“, kde linka se rovná proměnné „$hledej“
if (!$vysledek):                                           // pokud se dotaz nepovede
    echo "Doslo k chybe pri zpracovani dotazu v databazi.<BR>\n"; // tak vypiš zprávu
    break;                                                 // zastav skript
endif;
...

```

Konec PHP kódu

```
MySQL_Close($spojeni); // ukonči spojení s databází
} while (false);
?>
```

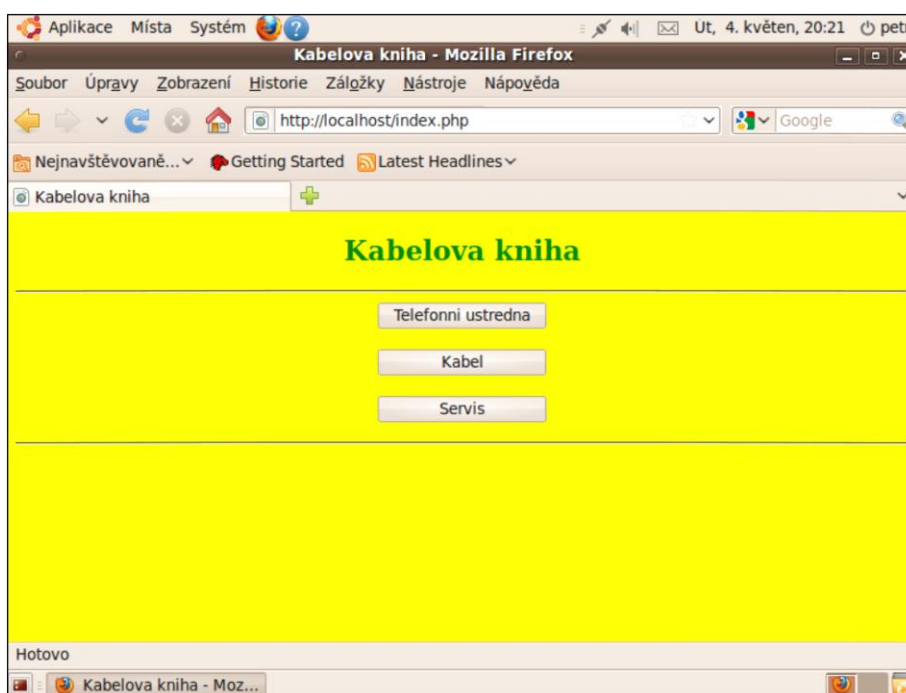
A končí opět HTML kódem

```
<FORM ACTION=tu.php METHOD=POST>
<INPUT TYPE=SUBMIT VALUE="Zpet do MENU TU" style="width: 150">
</FORM>
```

Postup byl následující. Nejdříve jsem vytvořil všechny stránky kabelové knihy, které jsem následně upravoval podle požadavku. Tím jsem vytvořil všechny formuláře, které obsahovaly převážně HTML kód. Celou kabelovou knihu můžeme rozdělit do tří částí. První část, je celá věnována telefonní ústředně, druhá část obsahuje vše ohledně kabelu a poslední část, je servisní.

12.5.2.1 Skripty určené pro část Telefonní ústředny

Do části, věnované telefonní ústředně se dostaneme z nabídky hlavní stránky, kde je MENU, které obsahuje tři tlačítka viz. obrázek 10. První tlačítko „Telefonní ústředna“, nás přesměruje do části pro telefonní ústřednu. Na hlavní stránce telefonní ústředny, se nacházejí tlačítka. Tlačítko „Nový záznam v TÚ“, slouží pro vložení nového telefonního čísla do tabulky ústředna v databázi kabelová kniha. Tlačítko „Vyhledat záznam“, nám umožňuje vyhledání HW pozice, telefonního čísla nebo typu v tabulce ústředna naší databáze. Posledním tlačítkem, „Přehled všech záznamu v TÚ“, vytvoříme tabulku, ve které budou obsaženy všechny záznamy z tabulky ústředny.



Obr. 11. Hlavní MENU webové aplikace „Kabelová kniha“

Zde uvádím komentovaný výpis ze skriptu „klapky.php“:

PHP zde převezme proměnnou poslanou z předchozí stránky za pomoci metody POST (pokud je v proměnné „setridit“ vložena hodnota) a vypíše jí do nadpisu, příkazem echo.

```
<TITLE>Vypis linek dle: <?PHP echo $_POST["setridit"]; ?></TITLE>
```

Dále jsem potřeboval u přepínačů typu „RADIO“, vyřešit problém při nastavení přepínače na výchozí hodnotu, při vybraném druhu seřazení u zvoleného přepínače. Použil jsem podobný způsob jako u nadpisu, přepínač se nastavil na výchozí hodnotu,

pokud se proměnná „setridit“ rovnala „HW_pozici“, „lince“ nebo „typu“, tak vypsál „CHECKED“ a tím se nastavil vybraný přepínač.

```
<INPUT TYPE="RADIO" NAME="setridit" VALUE="HW_pozice"  
<? if ($_POST['setridit']=="HW_pozice") echo ('CHECKED'); ?>>HW pozice  
<INPUT TYPE="RADIO" NAME="setridit" VALUE="linka"  
<? if ($_POST['setridit']=="linka") echo ('CHECKED'); ?>>Telefonního čísla  
<INPUT TYPE="RADIO" NAME="setridit" VALUE="typ"  
<? if ($_POST['setridit']=="typ") echo ('CHECKED'); ?>>Typu
```

Následuje část, kde obdobným způsobem výběru, jako u přepínačů „RADIO“, se provede SQL dotaz „SELECT“ v tabulce „ustredna“.

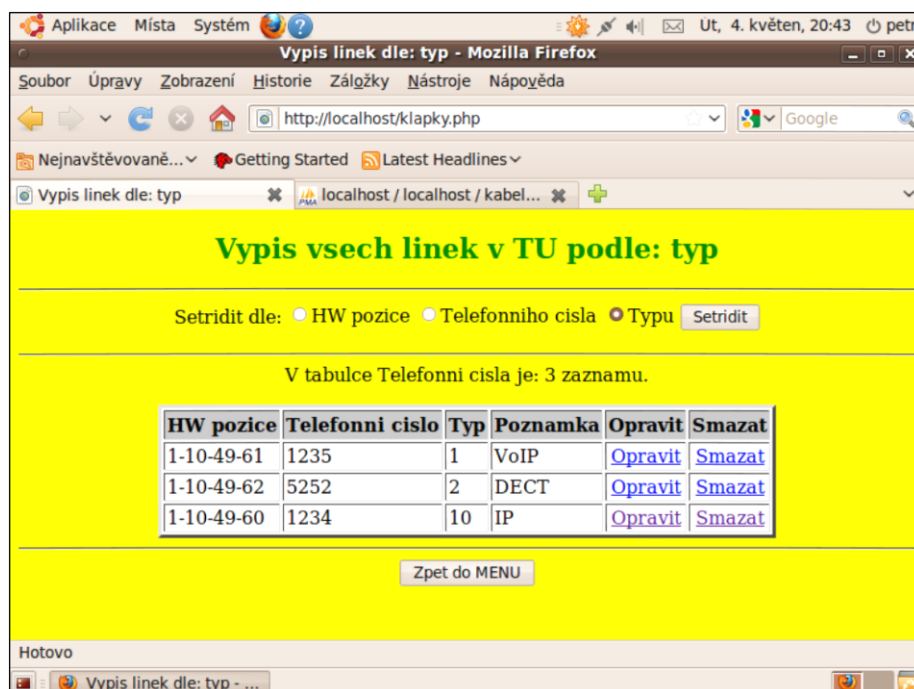
```
MySQL_Select_DB("kabelova_kniha"); // výběr databáze  
if($setridit == "HW_pozice"): // jestli se proměnná „setridit“, rovná „HW_pozici“, tak  
provede SQL dotaz „SELECT“, vybere záznamy z tabulky „ustredna“ a seřadí je dle  
„HW_pozice“  
@$setridit = MySQL_Query("SELECT * FROM ustredna ORDER BY HW_pozice");  
elseif($setridit == "linka"):  
@$setridit = MySQL_Query("SELECT * FROM ustredna ORDER BY linka");  
elseif($setridit == "typ"):  
@$setridit = MySQL_Query("SELECT * FROM ustredna ORDER BY typ,linka");  
elseif($setridit==""): // pokud je proměnná „setridit“ prázdná  
@$setridit=MySQL_Query("SELECT * FROM ustredna ORDER BY HW_pozice");  
  
else: // jinak provede příkaz „echo“ a vypíše ho na stránku  
echo "Doslo k chybě při zpracování dotazu v databázi.<BR>\n";  
Zjištění počtu záznamů v tabulce „ustredna“.  
  
echo "V tabulce Telefonní čísla je: ".MySQL_Num_Rows($setridit)." záznamů. ;  
  
Vypsání všech vybraných záznamů do tabulky. Cyklus „while“ se provádí pořád,  
pokud dostává hodnoty pro výpis.  
  
while ($zaznam = MySQL_Fetch_Array($setridit)):  
echo "<TD>".$zaznam ["HW_pozice"]."</TD>";  
echo "<TD>".$zaznam ["linka"]."</TD>";
```

```

echo "<TD>".$zaznam ["typ"]."</TD>";
echo "<TD>".$zaznam ["poznámka"]."&nbsp;</TD>";
Do tabulky jsem přidal odkazy, na opravu a smazání záznamu.
echo"<TD ALIGN=center>". "<A HREF='opravit_linku.php?Y_linka=$zaznam [linka]' >
Opravit</A></TD>";
echo"<TD ALIGN=center>". "<A HREF='smazat_linku.php?Y_linka=$zaznam[linka]'>
Smazat</A></TD>";

```

Do tabulky k vypsaným záznamům, jsou přidány odkazy, které nám umožní dále pracovat s daty v tabulce a můžeme je opravit nebo smazat. Výsledek skriptu „klapky.php“, vidíme na obrázku 11. Podrobnější popis funkcí všech stránek, je uveden v příloze „Příručka pro ovládání kabelové knihy“.



Obr. 11. Výpis záznamů z tabulky „ústředna“

Přehled všech skriptů, souvisejících s telefonní ústřednou:

tu.php – skript vytvoří MENU telefonní ústředny s přepínáním na další stránky

tu_novy.php – skript vytvoří formulář pro zadání nového telefonního čísla

pridat_tu.php – skript provede připojení do databáze, vloží vyplněná data a přesměruje na další stránku „klapky_new.php“

klapky_new.php - skript vypíše všechny vložená data do tabulky „ustredna“

tu_hledani.php – skript vytvoří formulář pro hledání v tabulce „ustredna“

klapky_linkax.php – skript vyhledá telefonní linku v tabulce „ustredna“ a vypíše ji do tabulky

klapky_HWx.php - skript vyhledá HW pozici v tabulce „ustredna“ a vypíše ji do tabulky

klapky_typx.php - skript vyhledá typ v tabulce „ustredna“ a vypíše ji do tabulky

klapky.php - skript vypíše všechny vložená data do tabulky „ustredna“ a umožní je třídit podle HW pozice, telefonního čísla a typu

opravit_linku.php – skript se připojí k databázi, vyhledá všechny údaje o telefonním čísle v tabulce „ustredna“ naší databáze a vypíše je do formuláře pro zadání nového telefonního čísla

pridat_linkux.php – skript se připojí k databázi, upraví všechny údaje v tabulce „ustredna“ a automaticky přesměruje na stránku „klapky.php“

smazat_linku.php – skript se připojí k databázi, vyhledá znovu všechny údaje o telefonním čísle v tabulce „ustredna“ a vypíše je do tabulky, zeptá se, zda opravdu chceme tento záznam vymazat s tabulky „ustredna“ naší databáze

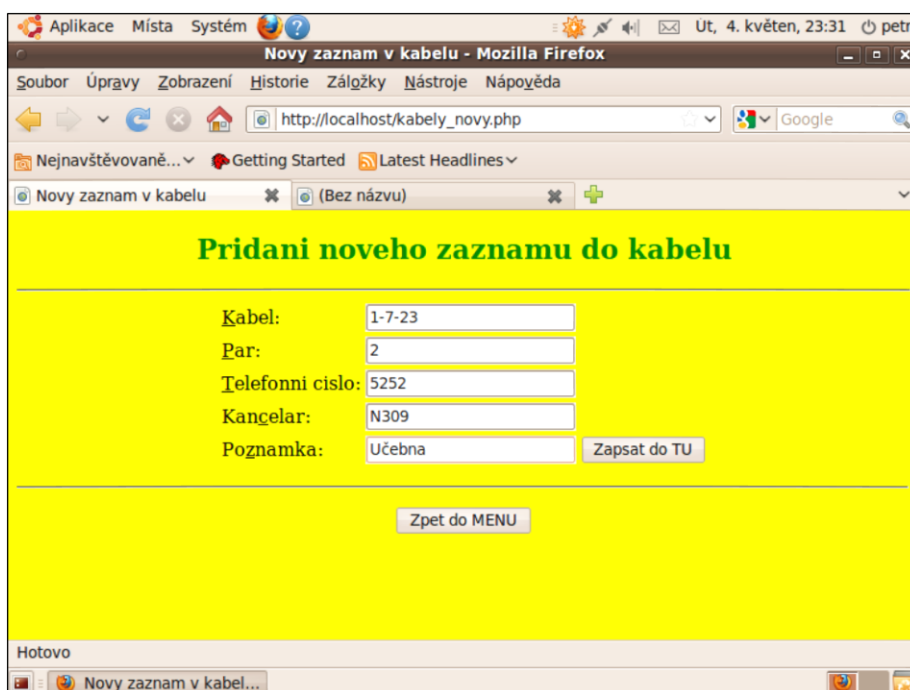
delete_linku.php - skript se připojí k databázi, vymaže záznam z tabulky „ustredna“ v naší databázi a vypíše zprávu o vymazání záznamu

opravit_linkux.php – skript se připojí k databázi, vyhledá údaje o záznamu, vloží je do formuláře pro přidání nového záznamu a požádá o jejich opravu

12.5.2.2 Skripty určené pro část kabel

Do části věnované kabelům se dostaneme z nabídky hlavní stránky, kde je MENU, které obsahuje tři tlačítka. Druhé tlačítko nás přesměruje na MENU kabelu. MENU obsahuje tlačítko „Zřídit nový kabel“, které nás přesměruje na formulář pro zadání nového kabelu. Další tlačítko nás přesměruje na stránku, kde se vypíše tabulka se všemi kabely, které máme v tabulce „kabely_popis“. Další tlačítko nás přesměruje na formulář, kde vyplňujeme údaje pro detail kabelu. Následující tlačítko nás přesměruje na stránku, kde se nachází formulář pro

vyhledávání v tabulce „kabely“ v naší databázi. Na obrázku 12 je zobrazen formulář na vložení nového záznamu do tabulky „kabely“, ze skriptu „kabely_novy“.



The screenshot shows a Mozilla Firefox browser window with the title "Novy zaznam v kabelu - Mozilla Firefox". The address bar shows "http://localhost/kabely_novy.php". The page has a yellow background and a green title "Pridani noveho zaznamu do kabelu". The form contains the following fields and values:

Label	Value
Kabel:	1-7-23
Par:	2
Telefonni cislo:	5252
Kancelar:	N309
Poznamka:	Učebna

There is a "Zapsat do TU" button next to the Poznamka field and a "Zpet do MENU" button at the bottom of the form.

Obr. 13. Formulář na vložení nového záznamu do tabulky „kabely“

Podrobnější popis funkcí všech stránek, je uveden v příloze „Příručka pro ovládání kabelové knihy“.

V aplikaci „kabelova kniha“ používám čtyři základní dotazy SQL, jako je:

SELECT – výběr záznamu

```
@$vysledek = MySQL_Query("SELECT * FROM kabely WHERE kabely_id=$X_linka");
```

UPDATE – oprava záznamu

```
$vysledek = MySQL_Query($spojeni="UPDATE `kabelova_kniha`.`kabely`  
SET `kabel` = '$kabel', `par` = $par, `linka` = $linka,  
`kancelar` = '$kancelar', `poznamka` = '$poznamka'  
WHERE `kabely`.`kabely_id` = $kabely_id LIMIT 1");
```

INSERT – vložení záznamu

```
@$vysledek = MySQL_Query($spojeni="INSERT INTO `kabelova_kniha`.`kabely_popis`  
(`kabel`, `nazev`, `pary`) VALUES ('$kabel', '$nazev', '$pary')");
```

DELETE – vymazání záznamu

```
$Y_linka = $_GET["Y_linka"];  
MySQL_Select_DB("kabelova_kniha"); // vyber databaze  
@$vysledek = MySQL_Query("DELETE FROM `ustredna` WHERE `linka`=$Y_linka  
LIMIT 1");
```

Přehled všech skriptů, souvisejících s kabely:

kabel.php – skript vytvoří MENU kabelu s přepínáním na další stránky

kab_novy.php – skript vytvoří formulář pro vyplnění dat, která chceme uložit do tabulky „kabely_popis“

pridat_kabel.php - skript provede připojení do databáze, vloží vyplněná data a přesměruje na další stránku „kabely.php“

kabely.php – skript provede připojení do databáze, najde všechny data v tabulce „kabely_popis“ a vypíše je do tabulky

kabely_novy.php – skript vytvoří formulář pro vložení nového záznamu, do tabulky „kabely“

opravit_kabelvse.php – skript se připojí k databázi, vyhledá data o kabelu, vytvoří formulář pro vložení nového kabelu a vypíše do nich vyhledaná data z řádku, který jsme chtěly opravit

pridat_kabelvse.php – skript provede připojení k databázi a opraví údaje, které jsme zadali, pak automaticky přesměruje na stránku „kabely.php“

smazat_kabelvse.php – skript se připojí do databáze znovu vypíše data, která chceme smazat a opět se zeptá, zda je chceme smazat

delete_kabelvse.php – skript provede připojení do databáze, smaže vybrané data a vypíše zprávu o jejich vymazání

kabel_hledani.php – vytvoří formulář pro zadání dat, které chceme vyhledat v tabulce „kabely“ naší databáze

kabel_kabelx.php – vyhledá všechny záznamy pro zadaný kabel, v tabulce „kabely“ a vypíše je do tabulky

kabel_linkax.php - vyhledá zadanou telefonní linku v tabulce „kabely“ naší databáze a vypíše jí do tabulky

kabel_kancelarx.php - vyhledá zadanou kancelář v tabulce „kabely“ naší databáze a vypíše jí do tabulky

pridat_kabely.php – skript převezme proměnné, které jsme vyplnili ve formuláři, připojí se k databázi, vloží data do tabulky „kabely“ a přesměruje na stránku „kabel.php“

pridat_kabelyx.php – skript převezme proměnné, připojí se k databázi, opraví data v tabulce „kabely“ a přesměruje na stránku „kabel_linkaxx.php“

kabel_linkaxx.php - skript se připojí k databázi, vyhledá záznam v tabulce „kabely“, vypíše data do tabulky a napíše zprávu, že záznam byl opraven

opravit_kabel.php – skript se připojí k databázi, vyhledá všechny údaje o kabelu v tabulce „kabely“ a vypíše je do formuláře pro vložení nového záznamu

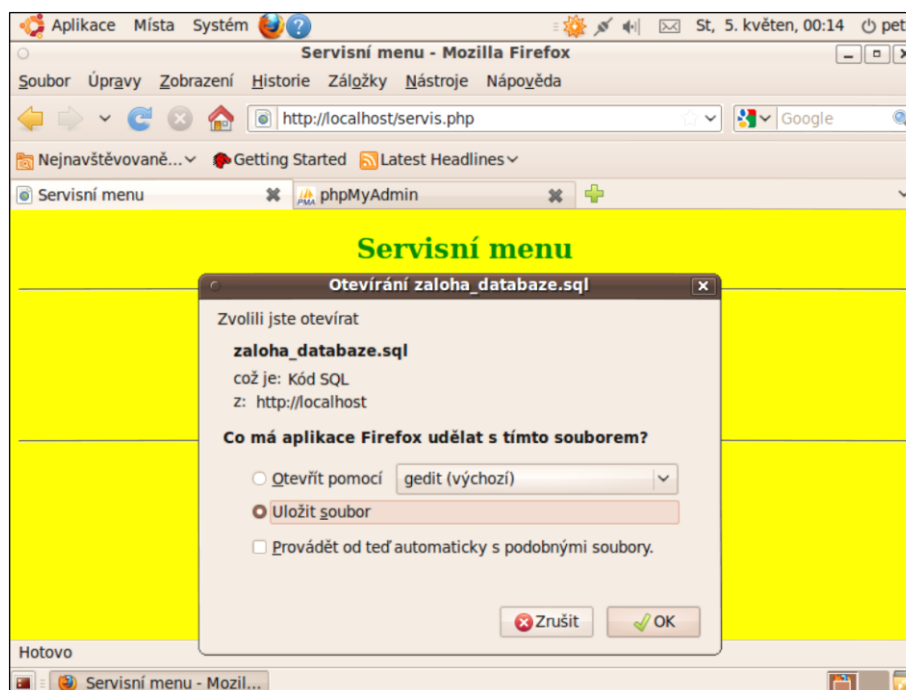
smazat_kabel.php – skript se připojí k databázi, vyhledá všechny údaje o vybraném záznamu, ty vypíše do tabulky a zeptá se, zda opravdu chceme tento záznam vymazat

delete_kabel.php – skript se připojí k databázi, vymaže zadaný údaj a vypíše zprávu o vymazání záznamu

12.5.2.3 Skripty určené pro část servis

Do části, věnované servisu se dostaneme z nabídky hlavní stránky, kde je MENU, které obsahuje tři tlačítka. Třetí tlačítko nás přesměruje na MENU servisu. Zde jsou tlačítka „Duplicita“, které nás přesměruje na stránku, kde se vypíšou duplicitní záznamy z tabulky „kabely“. Dále zde máme tlačítko „Správa hesel“, které nás přesměruje na stránku, kde můžeme zadat nového uživatele do tabulky „pripojeni“ nebo si zjistit jaké uživatele již v této tabulce máme. Posledním tlačítkem je „Záloha databáze“, provedeme zálohu celé databáze

„kabelová kniha“ do souboru s příponou sql a uložíme někam na náš počítač. V souboru jsou všechny SQL příkazy pro opětovné vytvoření všech tabulek a záznamy, které jsme tam měli uloženy. Vytvoření souboru zálohy „kabelové knihy“ viz. Obrázek 13. Podrobnější popis funkcí všech stránek, je uveden v příloze „Příručka pro ovládání kabelové knihy“.



Obr. 14. Vytvoření souboru se zálohou celé databáze „kabelova_kniha“

Přehled všech skriptů, souvisejících se servisem:

servis.php - skript vytvoří MENU servisu s přepínáním na další stránky

duplicita.php – skript se připojí do databáze, provede kontrolu duplicitních záznamu telefonní linky v tabulce „kabely“ a vypíše je do tabulky

uzivatel_novy.php – skript vytvoří formulář pro zadání údajů o novém uživateli databáze „kabelová kniha“

insert_heslo.php – skript se připojí do databáze, upraví heslo pomocí hashe MD5, uloží zadané údaje do tabulky „prihlaseni“ a přesměruje nás na další stránku „prihlaseni_vypis.php“

prihlaseni_vypis.php – skript se připojí k databázi, vyhledá všechny údaje v tabulce „prihlaseni“ a vypíše je do tabulky

prihlaseni_potvrzeni.php – skript se připojí k databázi, znovu vyhledá všechny údaje o zadaném uživateli a vypíše zprávu, zda opravdu chceme vymazat daného uživatele

prihlaseni_smazat.php – skript se připojí k databázi, vymaže zadaného uživatele a přesměruje na stránku „prihlaseni_vypis.php“

zaloha.php – vyvolá skript „mysqldump.php“, připojí se do databáze, vytvoří soubor „zaloha_databaze.sql“, zvětší limit paměti na 32MBa pošle mysqldump na výstup

mysqldump.php – tento jediný skript jsem převzal z internetu, skript udělá zálohu celé databáze do souboru „zaloha_databaze.sql“ a inicializuje otevření okna, pro výběr, kde chceme daný soubor uložit

12.5.2.4 Ostatní skripty

Zde jsou uvedeny skripty, které slouží pro autorizaci vstupu do webové aplikace

„kabelova kniha“ a hlavní MENU:

index.php – skript vytvoří hlavní MENU, s tlačítky

prihlaseni.php – skript vytvoří formulář pro přihlášení uživatele do webové aplikace „kabelová kniha“

overeni_prihlaseni.php – skript upraví údaje z formuláře „prihlaseni.php“, zejména heslo pomocí haše MD5, připojí se k databázi, zkontroluje zda se „jmeno“ a „heslo“ shodují z údaji v tabulce „prihlaseni“ a pokud ano přesměruje na stránku hlavního MENU webové aplikace „kabelové knihy“ nebo ne, tak zůstává ve formuláři

prihlaseni1.php – skript vytvoří formulář na autorizaci a napíše zprávu, že zadané „heslo“ nebo „jméno“ nebylo správné

Výpis, ze skriptu „overeni_prihlaseni“:

```
$jmeno = $_POST["jmeno");// inicializace proměnné „jmeno“
```

```
$jmeno=addslashes($jmeno);// vloží lomítka před problémové znaky
```

```
$heslo = $_POST["heslo"];
```

```

$x_jmeno = trim(strtolower($jmeno)); // upravíme udaje před kontrolou v tabulce

$x_heslo = md5($heslo);           // provedeme zakódování hesla pomocí hašovací
                                   funkce MD5

                                   // jmeno i heslo zároveň se musí zhodovat

$dotaz = MySQL_Query("SELECT * FROM `prihlaseni`
WHERE jmeno = '$x_jmeno' AND heslo = '$x_heslo'");

    // přesměrování zpět na přihlašovací formulář, pokud se přihlašovací údaje
    neshodují s uloženým záznamem v tabulce „prihlaseni“

if ($pocet==""):
Header("Location: http://localhost/prihlaseni1.php");

elseif ($pocet=="0"): //pokud zjistí, že se záznam v tabulce nenachází, tak vypíše echo
echo "V tabulce Hesel tento zaznam neni";

elseif ($pocet=="1"): //pokud se záznam nachází v tabulce „prihlaseni“, tak nás
    přesměruje na skript „index.php“, kde je hlavní MENU kabelové knihy

Header("Location: http://localhost/index.php");

```

Všechny vytvořené skripty vložíme do adresáře /var/www/ a do webového prohlížeče zadáme adresu <http://localhost/prihlaseni.php>. Otevře se nám stránka pro autorizaci přihlášení do webové aplikace „kabelová kniha“. Zadáme „jmeno“ a „heslo“, které jsme si předtím uložili do tabulky „prihlaseni“ a pokud jsme vše zadali správně, tak se dostaneme do úvodního MENU webové aplikace „kabelová kniha“. Můžeme začít používat novou webovou aplikaci a prostřednictvím formulářů, ukládat data do naší databáze.

13 Závěr

Cílem bakalářské práce bylo vytvoření funkční webové aplikace pro databázi kabelových rozvodů na VŠB-TU Ostrava. Při vytváření webové aplikace používat doporučený software, jako je operační systém LINUX, server Apache, databázový server MySQL a skriptovací jazyk PHP. Jako přílohu, vytvořit příručku pro ovládání webové aplikace.

V bakalářské práci, jsem se snažil vypsát stručný přehled o použitém softwaru a dalším dostupném softwaru, který se dá také použít při tvorbě webové aplikace. Nejsem programátor, takže vše jsem se snažil popisovat co nejjednodušším způsobem. Podařila se vytvořit funkční webová aplikace „Kabelová kniha“, která se dá ihned používat. Na přiloženém kompaktním disku, můžeme najít podrobnější návod („Příručka pro ovládání kabelové knihy“), jak se webová aplikace „Kabelová kniha“ používá. Vytvořená webová aplikace, může sloužit, jako základ pro vytvoření dokonalejšího databázového systému, na vedení kabelové dokumentace. Je zde prostor na vylepšení webové aplikace, jako například lepší zabezpečení, propracovanější formuláře, další výstupy seřazených dat z tabulek databáze, vytvoření tlačítek pro tisk uložených dat apod..

Během tvorby webové aplikace, jsem získal celkový přehled okolo práce s databázemi a zjistil velký rozsah této problematiky. Práce byla časově velmi náročná pro někoho, kdo se začíná věnovat práci ze skriptovacím jazykem PHP. Při tvorbě skriptu pro webovou aplikaci, bylo nutné často hledat, kde je chyba a nalézt řešení, jakým způsobem jí lokalizovat a následně odstranit. Většina použitého softwaru je open source, takže je přístupný všem, kteří mají zájem studovat a rozšiřovat tuto webovou aplikaci.

14 Seznam použité literatury

- [1] AMORE, Elizabet , et al. *Vytváříme webové aplikace v PHP5, MySQL a Apache*. Vydání první. Brno : Computer Press, a.s., 2006. 816 s. ISBN 80-251-1073-7, K1271.
- [2] Apache HTTP Server In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 14.1.2007, 7.3.2010 [cit. 2010-05-04]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Apache_HTTP_Server>.
- [3] BÍBR, Ivan. *Ubuntu9.10 CZ Praktická příručka uživatele Linuxu*. Vydání první. Brno : Computer Press, a.s., 2009. 287 s. ISBN 978-80-251-2826-8, K1765.
- [4] DELISLE, Marc. *PhpMyAdmin efektivní správa MySQL*. Vydání první. Brno : Zoner Press, 2004. 264 s. ISBN 80-86815-09-9, ZR417.
- [5] HRABÁNEK, Jan. *Site.the.cz* [online]. 12.12.2001 [cit. 2010-05-05]. Linux - Historie Linuxu. Dostupné z WWW: <<http://site.the.cz/index.php?id=23>>.
- [6] HTML In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9.3.2006, 30.4.2010 [cit. 2010-05-03]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/HTML>>.
- [7] LACKO, Luboslav. *SQL Hotová řešení*. Vydání první. Brno : Computer Press, a.s., 2003. 298 s. ISBN 80-7226-975-5, K0848.
- [8] MySQL In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 29. 11. 2006, 7. 4. 2010 [cit. 2010-05-03]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [9] PLAŠIL, Šimon. *Co je Linux?* [online]. 2004 [cit. 2010-05-05]. Historie. Dostupné z WWW: <<http://www.cojelinix.ic.cz/historie.html>>.
- [10] ZAJÍC, Petr. *Linuxsoft.cz* [online]. 27.5.2004 [cit. 2010-05-05]. PHP (1) - Historie a budoucnost. Dostupné z WWW: <http://www.linuxsoft.cz/article.php?id_article=171>.
- [11] ZAPLÉTAL, Lukáš. *Linuxexpres* [online]. 18.8.2006 [cit. 2010-05-05]. VMware 5.5. Dostupné z WWW: <<http://www.linuxexpres.cz/software/vmware-5-5?highlightWords=vmware>>.

15 Seznam obrázků

Obrázek 1. - Spuštění instalace VMware workstation	23
Obrázek 2. - Spuštění virtuálního počítače VMware workstation	24
Obrázek 3. - Instalace operačního systému Ubuntu do VMware workstation	25
Obrázek 4. - Pojmenování virtuálního počítače a zadání hesla	26
Obrázek 5. - Instalace LAMP serveru pomocí správce balíků programu Synaptic	27
Obrázek 6. - Výběr serveru Apache2 pro phpMyAdmin	29
Obrázek 6.a. - Výběr nastavení databáze pro phpMyAdmin	29
Obrázek 7. - Ověření, jestli nám pracuje Apache server	30
Obrázek 8. - Ověření, jestli nám pracuje phpMyAdmin	30
Obrázek 9. - Ověření, jestli nám funguje PHP	31
Obrázek 10. - Propojení tabulek databáze „kabelova_kniha“	34
Obrázek 11. - Hlavní MENU webové aplikace „Kabelová kniha“	36
Obrázek 12. - Výpis záznamů z tabulky „ustredna“	38
Obrázek 13. - Formulář na vložení nového záznamu do tabulky „kabely“	40
Obrázek 14. - Vytvoření souboru se zálohou celé databáze „kabelova_kniha“	43

16 Seznam příloh

Příloha 1. - CD